Complexity Reduction Schemes for Video Compression

by

Hamid Reza Tohidypour

B.Sc., Amirkabir University of Technology (Tehran Polytechnic), 2007
M.Sc., Amirkabir University of Technology (Tehran Polytechnic), 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical & Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December 2016

Abstract

With consumers having access to a plethora of video enabled devices, efficient transmission of video content with different quality levels and specifications has become essential. The primary way of achieving this task is using the simulcast approach, where different versions of the same video sequence are encoded and transmitted separately. This approach, however, requires significantly large amounts of bandwidth. Another solution is to use scalable Video Coding (SVC), where a single bitstream consists of a base layer (BL) and one or more enhancement layers (ELs). At the decoder side, based on bandwidth or type of application, the appropriate part of an SVC bit stream is used/decoded. While SVC enables delivery of different versions of the same video content within one bit stream at a reduced bitrate compared to simulcast approach, it significantly increases coding complexity. However, the redundancies introduced between the different versions of the same stream allow for complexity reduction, which in turn will result in simpler hardware and software implementation and facilitate the wide adoption of SVC. This thesis addresses complexity reduction for spatial scalability, SNR/Quality/Fidelity scalability, and multiview scalability for the High Efficiency Video Coding (HEVC) standard.

First, we propose a fast method for motion estimation of spatial scalability, followed by a probabilistic method for predicting block partitioning for the same scalability.

Next, we propose a content adaptive complexity reduction method, a mode prediction approach based on statistical studies, and a Bayesian based mode prediction method all for the quality scalability. An online-learning based mode prediction method is also proposed for quality scalability. For the same bitrate and quality, our methods outperform the original SVC approach by 39% for spatial scalability and by 45% for quality scalability.

Finally, we propose a content adaptive complexity reduction scheme and a Bayesian based mode prediction scheme. Then, an online-learning based complexity reduction scheme is proposed for 3D scalability, which incorporates the two other schemes. Results show that our methods reduce the complexity by approximately 23% compared to the original 3D approach for the same quality/bitrate. In summary, our methods can significantly reduce the complexity of SVC, enabling its market adoption.

Preface

This thesis presents research conducted by Hamid Reza Tohidypour under the guidance of Dr. Panos Nasiopoulos. A list of publications resulting from the work presented in this thesis and the rest of the work Tohidypour has done during his PhD are provided on the following pages and the Appendix.

The work presented in Chapter 2 has been published in [P1-P2]. The content of Chapter 3 has been published in [P3-P7]. A contribution was made to MPEG video compression standardization activities [P4]. Chapter 4 appears in [P8-P10]. Three other publications [P11-P13] are the other outcomes of Tohidypour's PhD work (see Appendix for more information).

The work presented in Chapters 2 and 4 of this thesis was performed by Hamid Reza Tohidypour, including algorithm designing, algorithm implementation, and manuscript writing. Hamid Reza Tohidypour was the main contributor for implementing the proposed algorithms, conducting the experiments, and analyzing the results. The works presented in [P8] were conducted with feedback from Dr. Victor Leung. Dr. Panos Nasiopoulos and Dr. Mahsa T. Pourazad provided guidance and editorial input into the manuscript writing.

The work presented in Chapter 3 was primarily performed by Hamid Reza Tohidypour, including designing and implementing the proposed methods, performing all experiments, analyzing the results, and writing the manuscripts. The works presented in [P5, P7] were conducted with suggestions and manuscript editing from Hossein Bashashati. For all the methods presented in Chapter 3, I received the guidance and editorial input of Dr. Panos Nasiopoulos and Dr. Mahsa T. Pourazad.

Work that has not been published yet or not directly related to the Thesis topic is not included in this manuscript. Over the course of his PhD, Hamid Reza Tohidypour has

participated in a number of other projects in collaboration with other colleagues. Publications resulting from these projects are listed in the Appendix [P11]-[P13]. The work presented in the above papers were conducted with the guidance and editorial input of Dr. Panos Nasiopoulos and Dr. Mahsa T. Pourazad. A complexity reduction method for mode search process in spatial scalability was also developed and the resulting paper will be submitted to an IEEE transactions journal.

This thesis was written by Hamid Reza Tohidypour, with editing assistance from Dr. Nasiopoulos.

List of Publications Based on Work Presented in This Thesis

[P1] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Adaptive Search Range Method for Spatial Scalable HEVC," in *Proc. IEEE International Conference on Consumer Electronics*, Las Vegas, USA, Jan. 2014.

[P2] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Probabilistic Approach for Predicting the Size of Coding Units in the Quad-tree Structure of the Quality and Spatial Scalable HEVC," *IEEE Transactions on Multimedia*, vol. 18, no. 2, pp. 182 – 195, Feb. 2016.

[P3] H. R. Tohidypour, M.T. Pourazad, and P. Nasiopoulos, "Content Adaptive Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC," in *Proc. 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013.

[P4] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Content Adaptive Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC," ISO/IECJTC1/SC29/WG11, JCTVC-L0042, MPEG Doc. m27368, Geneva, Switzerland, Jan. 2013.

[P5] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "An Encoder Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC", *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 664 – 674, Sep. 2016.

[P6] H. R. Tohidypour, H. Bashashati, M. T. Pourazad, and P. Nasiopoulos, "A Bayesian-based Fast Mode Assignment Approach for Quality Scalable Extension of the High Efficiency Video Coding (HEVC) Standard," in *Proc. 6th Conference Balkan Conference in Informatics (BCI 2013)*, Greece, Sep. 2013.

[P7] H. R. Tohidypour, H. Bashashati, M. T. Pourazad, and P. Nasiopoulos, "Online-learning Based Mode Prediction Method for Quality Scalable Extension of the High Efficiency

Video Coding (HEVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology* (Accepted), May 2016.

[P8] H. R. Tohidypour, M. T. Pourazad, P. Nasiopoulos, and V. Leung, "A Content Adaptive Complexity Reduction Scheme for HEVC-Based 3D Video Coding," in *Proc. 18th Conference on Digital Signal Processing (DSP 2013)*, Santorini, Greece, July 2013.

[P9] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "A Low Complexity Mode Decision Approach for HEVC-based 3D Video Coding Using a Bayesian Method," in *Proc. 39th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014.

[P10] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Online Learning-based Complexity Reduction Scheme for 3D-HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 10, pp. 1870 – 1883, Oct. 2016.

Table of Contents

List of Tables

List of Figures

List of Acronyms

2D	Two Dimensional
3D	Three Dimensional
3D-HEVC	3D High Efficiency Video Coding
3D-HTM	3D-HEVC Test Model
Acc	Accuracy
AMP	Asymmetric Motion Partitions
ASRM	Adaptive Search Range Method
AZB	All-Zero Block
AVC	Advanced Video Coding
BD-BR	Bjontegaard Delta Bit Rate
BD-PSNR	Bjontegaard Delta PSNR
BL	Base Layer
BV	Base View
BV_t	Base Texture View
CACRS	Content Adaptive Complexity Reduction Scheme
CTCs	Common Test Conditions
CTU	Coding tree unit
CU	Coding Unit
DCP	Disparity-compensated Prediction
DE	Disparity Estimation
DV	Dependent View

DV_t	Dependent Texture View
E_c	Current block in enhancement layer
E_L	Left block in Enhancement Layer
EL	Enhancement Layer
EMD	Early Merge Decision
E_T	Top Block in Enhancement Layer
ET	Early Termination
E_{TL}	Top Left block in Enhancement Layer
E_{TR}	Top Right block in Enhancement Layer
FMA	Fast Mode Assignment
GDV	Global Disparity Vector
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
HD	High Definition
IEC	International Electrotechnical Commission
ILR	Inter Layer Reference
ILRP	Inter Layer Reference Prediction
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JCT-VC	Joint Collaborative Team on Video Coding
JCT-3V	JCT on 3D Video Coding Extension Development
JVET	Joint Video Exploration Team

JVT	Joint Video Team
LCU	Large Coding Unit
LRC	Lowest Rate Distortion Cost
MAP	Maximum a Posteriori
MB	Macroblock
MCP	Motion-compensated Prediction
ME	Motion Estimation
MHC	Motion Homogeneity Category
MLE	Maximum Likelihood Estimation
MOS	Mean Opinion Score
MP	Mode Prediction
MPEG	Moving Picture Experts Group
MPI	Motion Parameter Inheritance
MV	Motion Vector
MVC	Multiview Video Coding
MVD	Multiview Video plus Depth
MVH	Motion Vector Homogeneity
NB-FMA	Naive Bayes FMA
PCPM	Proposed CTU Prediction Method
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QP	Quantization Parameter
QPB	Quantization Parameter of the BL

QPE	Quantization Parameter of the EL
QPD	Quantization Parameter of the Depth
QPV	Quantization Parameter of the View
RA	Random Access
RA-HE	Random Access High Efficiency
RD	Rate Distortion
RDOQ	Rate Distortion Optimized Quantization
SAO	Sample Adaptive Offset
SDC	Simplified Depth Coding
SDR	Standard Dynamic Range
SHM	Scalable HEVC Test Model
SHVC	Scalable HEVC
SNR	Signal to Noise Ratio
SR	Search Range
SVC	Scalable Video Coding
Synth	Synthesized
TR	Time Reduction
TU	Transform Units
UHD	Ultra High Definition
VCEG	Video Coding Experts Group

Acknowledgments

This thesis would never have been done without the help and support from numerous people. Needless to say, I thank all of them. In particular, I would like to take this opportunity to express my thankfulness to the following individuals.

First and foremost, I would like to give my sincerest gratitude to my supervisor Dr. Panos Nasiopoulos for his continuous guidance and support throughout my Ph.D. studies. I thank Dr. Nasiopoulos for his encouragement, patience, enthusiasm, and broad knowledge in multimedia. I also thank Dr. Nasiopoulos for the freedom and independence he gave me during PhD research. He has always been a great mentor, a role model, and a friend.

I would like to extend my sincere thanks to Dr. Mahsa T. Pourazad for her support and guidance during my PhD studies. I thank Dr. Pourazad for her patience, encouragement, and great inspiring suggestions. I also want to thank Hossein Bashashati for his knowledge in the area of machine learning. I was lucky to have the opportunities to collaborate with these brilliant researchers. I also thank my other colleagues at the UBC Digital Multimedia Lab, Dr. Lino Coria, Dr. Di Xu, Mohsen Amiri, Dr. Bambang Ali Sarif, Sima Valizadeh, Maryam Azimi, Dr. Amin Banitalebi, Dr. Zicong Mai, Dr. Ronan Boitard, Dr. Colin Doutre, Basak Oztas, Stelios Ploumis, Cara Dong, Pedram Mohammadi, Ilya Ganelin, Fujun Xie, Ahmad Khaldieh, Timothee Bronner, Abrar Wafa, Iliya Koreshev, and Anahita Shojaei. It was a pleasure working with you all in such a vibrant and eclectic environment.

Next, I would like to express my gratitude to the following individuals, who have kindly helped me in one way or another during my Ph.D. studies: Dr. Victor Leung, Dr. Z. Jane Wang, Dr. Rabab Ward, Dr. Edmond Cretu, Dr. Shahriar Mirabbasi, Dr. Alan Wagner, Dr. Lawrence Walker, Dr. Sarbjit Sarkaria, Dr. Vikram Krishnamurthy, Dr. Jim Little, Dr. Ehsan Vahedi, Dr.

Ehsan Nezhadarya, Dr. Amir Valizadeh, Dr. Mani Malek Esmaeili, Dr. Tanaya Guha, Dr. Davood Karimi, Hamid Palangi, Simon Fauvel, Fahimeh Sheikhzadeh, Hiba Shahid, Hesham Mahrous, and Dr. Angshul Majumdar.

I would like to thank the UBC Graduate Studies for financially supporting me with the UBC Support Initiative (GSI) Awards.

Finally, I thank my parents and my beloved brother for their constant love, support and encouragement.

Dedication

To my parents and

my beloved

brother

1. Introduction and Overview

Using video applications on a variety of devices has become interwoven into our everyday lives. This is mainly due to the availability of a wide range of video-enabled gadgets and mobile devices with network connectivity. To transmit video content to heterogeneous devices, this content needs to be encoded in a way that is compatible with the playback capabilities of the specific device. To support all kind of display devices, one solution is to encode the video content with several configurations (compatible with devices) and transmit them separately (simulcast coding). This approach is computationally expensive and requires large amounts of bandwidth. Another solution is to use Scalable Video Coding (SVC), which enables multicast service and video transmission to heterogeneous clients with different capabilities [1], [2]. An SVC stream consists of a base layer (BL) and one or more enhancement layers (ELs). On the decoder side, based on the type of the application and supported complexity level, the appropriate part of an SVC bit stream will be decoded. Depending on the specifications of the device and the limitations of the network, different types of scalabilities including SNR/Quality/Fidelity, spatial (resolution), temporal (frame rate), color bit depth (low dynamic range and high dynamic range), and the number of views (2D and 3D) or a combination of these scalabilities may be used [3]. However, the redundancies introduced by scalable coding between the different versions of the same stream allow for complexity reduction, which in turn will result in simpler hardware and software implementation and facilitate the wide adoption of SVC. As similar task and opportunity for complexity reduction arises in the case of multiview applications, where two or more views share considerable amount of information compared to a single view coding.

Several years back, scalable coding was introduced in the form of H.264/SVC, which was an extension of the H.264/AVC standard [1]. H.264/SVC supports temporal, spatial, SNR/Quality/Fidelity scalabilities as well as combined scalability (a combination of the temporal, spatial and SNR scalabilities).

Recently, with the standardization of the High Efficiency Video Coding (HEVC), which has substantially higher compression capabilities (up to 45.54% in terms of bit rate) than the H.264/AVC standard [4], [5] and the renewed interest from industry in scalable coding, the Joint Collaborative Team on Video Coding (JCT-VC) of Moving Pictures Experts Group (MPEG) and Video Coding Experts Group (VCEG) of ITU-T introduced the scalable extension of HEVC, known as SHVC [6], [7].

Considering the superior performance of HEVC and the market trend towards the adoption of a multiview system, the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of the ISO/IEC MPEG and the ITU-T have developed the 3D extension of HEVC (3D-HEVC) with the objective to provide efficient compression of multiview video sequences [8,9].

In this thesis we address complexity reduction for spatial scalability, SNR/Quality/Fidelity scalability, and multiview scalability for the HEVC standard. Reducing the complexity of these standards allows for a simpler hardware and software implementation of the above standards and that in turn will facilitate their wide adoption in the market. All our methods are implemented in the MPEG reference software (SHM and 3D-HTM) and our results have been shared with industry through MPEG contributions.

In Chapter 2, we present two methods for reducing the complexity of the spatial extension of HEVC. Chapter 3 deals with the reduction of the computational complexity of the quality

extension of HEVC. In Chapter 4, we present three schemes for the complexity reduction of 3D-HEVC.

The following Sections in this introductory Chapter provide background information on HEVC, SHVC, and 3D-HEVC. They also provide a literature review of the previously proposed complexity reduction methods for those standards. Subsection 1.1.1 includes short overview of the HEVC standard. Subsection 1.1.1.2 elaborates on the existing complexity reduction methods proposed for HEVC. Subsection 1.1.2 provides basic background information of scalable video coding, the scalable extension of H.264 (H.264/SVC), and the SHVC standard. Existing works dealing with complexity reduction of H.264/SVC and SHVC standard are reviewed in Subsection 1.1.2.1.1 and Subsection 1.1.2.2.1. Subsection 1.1.3 introduces Multiview/3D video coding and the multiview video coding (MVC) standard. Subsection 1.1.3.1 elaborates on the existing complexity reduction methods proposed for MVC. Subsection 1.1.3.2 provides background information on 3D-HEVC. A literature review on the complexity reduction of 3D-HEVC is presented in Subsection 1.1.3.2.3. Section 1.2 concludes the introduction with an overview of the research contributions presented in this thesis.

1.1 Overview of Existing Work and Literature Review

In this Section we give an overview of HEVC, H.264/SVC, SHVC, MVC, and 3D-HEVC. In addition, we review existing works for complexity reduction of the above-mentioned standards.

1.1.1 Overview of High Efficiency Video Coding

The HEVC standard is one of the most recent joint video projects of the ITU-T VCEG and the ISO/IEC MPEG standardization organizations, which was finalized in January 2013. HEVC

offers a substantially higher compression performance compared to the previous major video coding standard (H.264/AVC [10]). Objective comparison results show that the current HEVC design outperforms H.264/AVC by 29.14% to 45.54% in terms of bit rate or 1.4 dB to 1.87dB in terms of PSNR [4]. The subjective comparison of the quality of compressed videos – for the same (linearly interpolated) Mean Opinion Score (MOS) points shows that HEVC outperforms H.264/AVC, yielding average bitrate savings of 58% [11].

HEVC utilizes a quad-tree based coding structure with support for coding units of more diverse sizes than that of macro-blocks in H.264/AVC. The basic block in HEVC is known as the coding tree unit (CTU), whose size is usually set to 64×64. CTU, which is also referred to as largest coding unit (LCU), can be recursively split into smaller Coding Units (CU), which in turn can be split into small Prediction Units (PU) and Transform Units (TU). Note that the process of splitting CTU into smaller CUs is continued for D iterations. Here, D indicates the maximum CU depth in the quad-tree structure of the CTUs [12]–[14] .

To reduce the spatial and temporal redundancies of video frames, HEVC employs more complicated intra prediction modes and more flexible motion compensation than H.264/AVC [7]. For intra prediction, HEVC uses 35 luma intra prediction modes compared to 9 used in H.264/AVC. Furthermore, intra prediction can be done at different block sizes, ranging from 4×4 to 64×64 (depending on the size of the PU).

In the case of inter-prediction, HEVC introduces a technique called “motion merge”. For every inter-coded PU, the encoder can choose between 1) the motion merge mode, 2) the SKIP mode, or 3) explicit encoding of motion parameters. The motion merge mode involves creating a list of previously coded neighboring (spatially or temporally) PUs (called candidates) for the PU being encoded. The motion information for the current PU is copied from one selected candidate,

avoiding the need to encode a motion vector for the PU; instead HEVC encodes only the index of a candidate in the motion merge list as well as the residual data. In the SKIP mode, the encoder signals the index of a motion merge candidate and the motion parameters for the current PU are copied from the selected candidate. However, unlike the motion merge mode, for the Skip mode, the encoder does not send any residual data. This allows areas of the picture that change very little between frames to be encoded using very few bits.

In explicit coding, inter-coded CUs can use Symmetric and Asymmetric Motion Partitions (AMP). AMPs allow for asymmetrical splitting of a CU into smaller PUs. AMP can be used on CUs of size 64x64 down to 16x16, improving coding efficiency since it allows PUs to more accurately conform to the shape of objects, without requiring further splitting [12]–[14]. For each inter-prediction coded PU, the HEVC encodes a set of motion parameters, which consists of a motion vector, a reference picture index and a reference list flag.

Although the increased number of intra modes, and more flexible inter prediction improve the coding performance of HEVC, they also result in increased computational complexity. For each mode, the HEVC encoder should perform transform, quantization, entropy coding, inverse quantization, inverse transform, and pixel reconstruction to compute the accurate rate distortion cost. The complexity of HEVC's encoder has been an important research subject from the early stages of its designing [15]–[17].

1.1.1.1 Overview of Complexity Reduction Methods Designed for HEVC

To reduce the complexity of the HEVC encoder, several methods have been proposed [18]–[44]. To facilitate intra prediction, a fast mode decision method is proposed in [18]. In [19], researchers proposed an early coding unit splitting and pruning method for intra coding. In

another study, a low-complexity algorithm based on level and mode filtering that reduces the angular modes is proposed for HEVC Intra prediction [20]. In [21], a texture complexity based fast prediction unit size selection algorithm is proposed for HEVC Intra coding. Researchers in [22] propose a low complexity HEVC Intra coding method for high-quality mobile video communication which predicts the most probable depth range based on the spatial correlation among CUs. Then, in order to improve the prediction accuracy, a statistical model-based CU decision approach is proposed in which adaptive early termination thresholds are determined and updated based on the rate distortion (RD) cost distribution, video content, and quantization parameters. In [23], a complexity reduction method is proposed for HEVC Ultra high definition formats (UHD) coding which predicts coding modes and quad-tree structure from those optimized for the lower (high definition (HD)) resolution version of the input UHD video. In another study, researchers propose a low-complexity block size decision for HEVC intra coding using binary image feature descriptors [24]. In [25], a CU size selection method is proposed for intra prediction, which uses gradient information of the CU image segment to derive texture complexity.

As in the low-delay and random-access configurations of the HEVC encoder, inter prediction is the major time-consuming process, several studies have focused on decreasing the complexity of inter prediction [26]–[32]. In [26], an effective CU size decision method is proposed that uses two approaches to reduce the complexity of HEVC encoder. The first approach determines the quad-tree depth level. The second approach reduces the complexity of the motion estimation for small block sizes. In [28], an early merge mode detection method is suggested that uses the mode of the root blocks, all-zero blocks, and the motion estimation information. In [31], a fast CU selection method is proposed which uses motion divergence to choose the CU size. A coding tree

depth estimation method is presented in [32] which uses the spatial and temporal correlations for complexity reduction of HEVC. In another study, a fast PU decision algorithm is proposed for HEVC, which uses the correlation among spatial-temporal PU modes and texture complexity [33]. In [34], a fast CU partitioning method is proposed for HEVC which uses Bayesian classifier for prediction. Researchers in [43] propose a method that uses motion homogeneity and RD cost information for CU size prediction.

1.1.2 Overview of Scalability and Scalable Video Coding

In general, a scalable video bitstream is a stream in which either some parts or the entire stream can be decoded to match the capabilities of the display device. Spatial scalability is a type of scalability in which the bit-stream can be pruned to some decodable subsets, which represent the original content at different resolutions. On the other hand, temporal scalability describes cases in which the resulting subsets represent the source content at different frame rates. Quality scalability is a term used to explain cases in which the sub-bitstreams provide the same spatio-temporal resolution as the original bitstream with different visual qualities. Quality scalability is also known as fidelity or signal to noise ratio (SNR) scalability [1].

1.1.2.1 Overview of the Scalable Extension of H.264

The previous scalable video coding standard, known as H.264/SVC, is an extension of the H.264/AVC standard [1]. H.264/SVC supports temporal, spatial, SNR/Quality/Fidelity scalabilities as well as combined scalability (a combination of the temporal, spatial and SNR scalabilities).

In H.264/SVC there are three inter-layer prediction schemes: Inter-layer texture prediction, inter-layer motion prediction, and inter-layer residual prediction. In residual prediction, the residual signal of the corresponding block in the lower layer is up-sampled block-wise (if necessary) and the resulting signal is used as the reference to predict the residual signal of the to-be-encoded block in the enhancement layer. Hence, the difference between the prediction signal and residual signal is encoded instead of the residual signal of the enhancement layer [1].

In Inter-layer motion prediction, the partitioning data of the EL block together with the associated reference indexes and motion vectors are derived from the corresponding coding information of the co-located motion vectors of the corresponding blocks in the reference layers. Note that inter-layer motion prediction is available for the case that the corresponding block in BL is inter-coded. In Inter-layer texture prediction, a reconstructed signal of a corresponding block in lower layer is up-sampled (if necessary) and the resulting signal is used for predicting the to-be-encoded block in the enhancement layer [1].

1.1.2.1.1 Overview of the Complexity Reduction Methods Designed for H.264/SVC

The scalability features explained in the previous Subsection come along with a significant increase in coding complexity compared to H.264/AVC. The complexity of H.264/SVC is mainly due to the additional temporal and spatial prediction processes involved in coding the multiple layers. Reduction of the scalable encoder's complexity is required especially for real-time applications, where processing power is limited. This has been achieved by utilizing the correlation between the base layer and enhancement layers. In [45]–[57], the coding information (i.e., mode) of the already encoded neighboring blocks in the base and/or enhancement layer is utilized to reduce the computational complexity of H.264/SVC. In [45], the complexity of temporal prediction is reduced by computing the conditional probability that a specific inter

prediction mode is selected for encoding a macroblock (MB) in the enhancement layer given the mode of the corresponding MB in the base layer. This statistical information is used to select the appropriate inter prediction mode for encoding MBs in the enhancement layer, based on the selected modes for the corresponding MBs in the base layer and neighboring MBs (Top and Left MBs) in EL. The method is employed to decrease the complexity of quality and spatial scalability. In [46], [54], an early mode-search termination (ET) method was proposed, which uses the RD cost values of the already encoded blocks to predict the RD cost of to-be-encoded blocks in the enhancement layer for the quality scalability. The proposed ET method is only applied to the skip and inter-prediction modes of H.264/SVC. To terminate the EL mode search of SVC with spatial scalability, researchers in [47] propose to use an all-zero block (AZB) detection method. In [55], a threshold for RD cost is proposed for mode prediction in the EL. In [56], a low complexity mode decision method for spatial scalability is used for predicting the zero motion and zero transform coefficients in the current to-be-coded block of EL uses the coding information of the corresponding block in the lower layer. In [57], it is reported that the Intra16*16 mode is not required to check for encoding the EL. So, to decrease the complexity of spatial scalability few modes will be checked for intra prediction when the corresponding BL is encode using intra prediction [57]. In addition, a fast mode decision method is suggested to decrease the complexity of temporal scalability. Each MB is classified into two different classes, namely, MBs with high motion and MBs with low motion. When the corresponding block in the base layer belongs to MB with low motion, its MB mode will be checked for the to-be-encoded MB in EL. Otherwise, the MB modes of the already encoded neighboring blocks (upper block and left block) will be checked. Finally, the RD cost of the skip mode is also computed. The chosen MB mode is the mode which has the lowest RD cost [57].

1.1.2.2 Overview of Scalable Extension of HEVC

HEVC's compression performance has led to a significant interest from industry in developing a scalable version of this standard. To address this interest, MPEG and ITU have introduced the scalable extension of HEVC, known as SHVC [58]–[60]. In this regard, high-level syntax design of existing HEVC standard was modified to provide SHVC which supports quality scalability, spatial scalability, and their combinations. In order to improve the compression performance, SHVC uses inter-layer motion prediction and inter-layer texture prediction, in addition to the advanced prediction features of HEVC [58]. In the inter-layer motion prediction process of SHVC, in addition to the temporal and spatial neighboring PUs currently used in HEVC, the motion parameters (motion vector and the reference picture index) of the co-located CU in the BL can be added to the merge candidate list [58]. In the Inter-layer texture prediction of SHVC, inter-layer reference pictures (or its up-sampled version) are used as the reference pictures in addition to the temporal reference pictures [58]. To enable this prediction mode in HEVC, there is no need to change the block level syntax (which determines the CU splits) and decoding process. Here, only the high level syntax has been changed to add the corresponding reconstructed pictures from the reference layers to the reference list. In this prediction mode, inter-layer reference pictures are used as the reference pictures in addition to the temporal reference pictures. Moreover, to specify whether the current prediction Unit (PU) is predicted from an inter-layer reference picture or a temporal reference picture, a signaled reference picture index is used [58].

Considering that HEVC is already complex, it is inevitable that the scalable extension with its multi-layers of scalability to also be highly complex. Reduction of this complexity, which will

address cost efficiency and power requirements, is one key factor that may enable the widespread adoption of this emerging standard.

1.1.2.2.1 Overview of the Complexity Reduction Methods Designed for SHVC

Some work has been done to address the complexity of SHVC [61]–[63]. A fast mode decision method is proposed in [63] for all intra spatial scalability. This approach uses the correlation of the CU depth and intra prediction modes of the enhancement layer and the base layer for mode prediction. In [61], a low-complexity intra coding tool has been proposed to the scalable HEVC call for proposals based on content statistics. In [62], a fast mode and depth decision algorithm for Intra prediction of Quality SHVC is proposed. This method only checks partial modes based on the relationship between the modes. In addition, it skips the depths with low possibilities that are determined based on their inter-layer correlations and textural features. As mentioned before, the methods proposed in [61]–[63] are only able to reduce the complexity of intra coding in SHVC but fail to address the complexity in inter prediction for spatial SHVC.

1.1.3 Introduction to Multiview Video Coding

Multiview display systems are anticipated to be available in the consumer market at affordable prices in the next few years. This projection stems from the recent advances in display technology, the showcase of high-quality multiview display prototypes in recent trade shows and the ultimate goal - glasses free 3D TV. This trend and evolution have become the driving force for industry to seek new solutions that will enable the end-to-end multiview pipeline.

One of the major challenges in implementing multiview services is the compression and transmission of multiview content, which includes several simultaneous video sequences from

the same scene. To address this issue, the MVC standard was developed by the Joint Video Team (JVT) of the ISO/IEC MPEG and the ITU-T VCEG [64]–[66]. MVC is an amendment to the H.264/MPEG-4 AVC video compression standard that enables efficient encoding of sequences captured simultaneously from multiple cameras by taking advantage of the correlation between the different views as well as the spatial and temporal correlation within the frames of each view. In MVC, one of the views is encoded using the conventional H.264/AVC. For coding the other views, in addition to previously encoded pictures of the same view, already encoded pictures of the other views can be used as reference pictures. These pictures are added to the reference list for the motion prediction. Conventional stereo displays need only two views. Autostereoscopic displays, on the other hand, need many more views in order to produce the immersive multiview feeling.

To avoid sending several views, we can transmit 3D content in the format known as Multiview Video plus Depth (MVD) where a few views are accompanied by their corresponding depth data. At the decoder side, the transmitted views and the depth maps are used for synthesizing additional views to address the needs of the specific autostereoscopic display.

Recently, JVT (MPEG+VCEG) has developed the 3D extension of HEVC (3D-HEVC) to provide efficient compression for the 3D content [8]. Section 1.1.3.1 includes the complexity reduction methods proposed for MVC. Section 1.1.3.2 gives a brief overview of the 3D-HEVC codec and the existing complexity reduction methods proposed for 3D-HEVC.

1.1.3.1 Overview of the Complexity Reduction Methods Proposed for MVC

The coding complexity of MVC is significantly higher than that of H.264/AVC, due to the greater number of views needed to be encoded, making the adoption of MVC for real-time

applications challenging. Several methods have been proposed for reducing the complexity of MVC [65]–[70]. For instance, in [63], a view adaptive motion search method is proposed that decreases the complexity of the motion search process based on the motion homogeneity of the matching block in the reference view using the global disparity vector (GDV). This method suggests search range values based on statistical studies done on MVC. Then, the complexity of the mode search process is decreased by examining only a limited number of prediction modes, based on information from the corresponding block in the reference view and its eight spatial neighboring blocks [63]. Another study suggests classifying each MB of a video frame into a near region, middle region or far region based on the depth map of the scene, if depth information is available (multiview plus depth) [64]. Then based on the relations between depth information and the MBs' prediction mode, the mode-search is limited to a small number of modes. In [67], based on the motion homogeneity a fast motion and disparity estimation method is proposed. The study in [68] proposes a method to reduce the complexity of the mode-search process, so that the encoder does not check all the modes for finding the one with the lowest rate RD cost. This study suggests that if the corresponding block in the adjacent view as well as its eight available neighboring blocks are encoded as Skip mode (these nine blocks are called predictor blocks), then the current to-be-code block will also be encoded as Skip mode. Otherwise, the encoder uses an early mode-search termination approach, which significantly reduces the search complexity. Researchers in [69], propose an iterative search method for the motion and disparity estimations that significantly reduces the complexity of stereoscopic video coding compared to the full search algorithm. In [70], based on the relations between the motion and disparity vectors, a fast disparity estimation and motion estimation method is proposed.

1.1.3.2 Overview of the 3D - High Efficiency Video Coding

The 3D-HEVC standard involves coding a limited number of views and the corresponding depth maps. To support the variety of multiview display systems, this limited number of views and their depth information are used to synthesize several additional views at the decoder side. Compared to the traditional format, where multiview content includes a large number of views (without depth map), this format simplifies capturing by limiting the number of views (e.g., 2 or 3) and at the same time reduces bandwidth requirements.

3D-HEVC utilizes all the advanced features of HEVC, including flexible partitioning quad-tree structure and sophisticated and diverse inter and intra prediction modes (3 inter-prediction modes and 35 intra prediction modes).

3D-HEVC also uses a quad-tree based coding structure with support for diverse block sizes. Similar to HEVC, the basic block in 3D-HEVC is known as the CTU. In the quad-tree structure, each CTU is divided into four equal-sized CUs. Each CU is then divided into four smaller CUs. The process of splitting CTU into smaller CUs is continued for D iterations. Here D indicates the maximum CU depth in the quad-tree structure. For instance if the CTU is not divided into sub-blocks, the depth (D) is equal to zero, and if the CTU is divided into four CUs, then the depth is equal to one. The CUs can later be split into small PUs and TUs [5].

Fig. 1.1 shows the basic structure of the 3D-HEVC codec [71]. All the views and depth maps, which belong to the video scene for the same time instance, make an access unit. The access units are encoded consecutively. As it can be observed, in each access unit, one of the views of the multiview sequence (called the base view) and corresponding depth map are encoded using the unmodified HEVC independently from the other views and their depth maps.

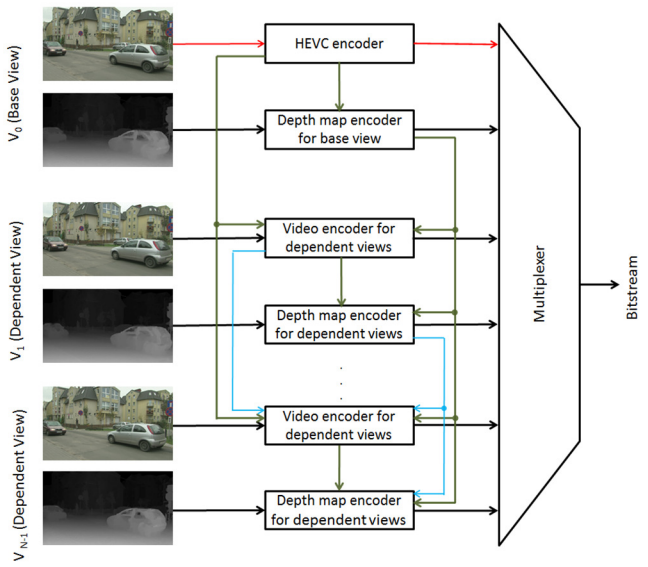

Figure 1.1 The structure of 3D-HEVC.

1.1.3.2.1 Coding of the Dependent Texture Views

For the dependent views, 3D-HEVC utilizes disparity-compensated prediction (DCP) in addition to the spatial prediction and motion-compensated prediction (MCP) used in HEVC. In DCP, the already encoded frames of other views in the same time instance are used for prediction (see arrows in Fig. 1.1). This tool improves the coding efficiency. To find the efficient tool for each to-be-encoded block, the encoder compares MCP to DCP. In this competition the MCP usually is chosen, meaning that DCP is only used for a small part of the to-be-encoded frame [72].

Two additional inter-view prediction methods have been added to increase the compression efficiency of MCP. Inter-view motion prediction is a new technique in 3D-HEVC, which intends to take advantage of the high correlation between the motion vectors of the different views at the same time instance. In this case, the motion parameters of the corresponding CUs in the reference views are added to the motion candidate list for the to-be-encoded CU [72]. Here, the

corresponding CUs in the reference view are found using the depth information. This method provides some candidates, which will be added to the candidate list of the so-called merge mode in HEVC.

The second additional inter-view prediction method is known as inter-view residual prediction. In this method, the reconstructed residual signal of an already encoded picture of the reference view in the same access unit (same time instant) can be used to provide efficient compression. Here, the corresponding block in the reference view is estimated by using the disparity vector from the depth information. Then, the residual signal of the to-be-encoded block is subtracted from the corresponding block in a reference view. If a block uses an inter-view residual prediction, only the resulting difference signal is transformed and encoded [72].

1.1.3.2.2 Coding of Depth Maps

Depth map streams have different texture characteristics compared to regular videos, and, thus, the existing HEVC intra-prediction modes do not effectively compress them. Some of the common characteristics of depth map frames include large areas with approximately constant or slowly varying values and sharp edges when depth level changes. 3D-HEVC is equipped with the prediction modes specially designed for compressing depth maps. In these modes, a depth block is approximated by a model that splits depth into two different non-rectangular regions using either a straight line (Wedgelets partitioning) or a complex geometrical curve (Contour partitioning) [72]. Then, each of these two regions is represented by a constant value. The first depth modeling mode is called Explicit Wedgelet. In this mode, the best Wedeglet partitioning is found for each depth block by searching over a set of Wedgelet partitions. During this search process, the Wedgelet partition that leads to the minimum distortion between the original block and the approximated block is selected. The second mode is called Restricted signaling and inter-

component prediction of Wedgelet partitions [9]. The last mode is called Inter-component prediction of Contour partitions and uses the reconstructed co-located block in the associated video frame as a reference to predict the Contour partitioning of a depth block [9]. Considering that each texture view and its corresponding depth map illustrate the dynamics of the same scene, they both have similar motion characteristics. Thus, in 3D-HEVC the current to-be-coded CU in the depth stream inherits the partitioning and associated motion vector information of the co-located CU in the corresponding view. In this regard, the existing merge and skip modes in HEVC have been modified for signaling the motion parameter inheritance (MPI) and the list of merge candidates has been extended for using the motion information of the co-located block in the video stream [9].

Since in the 3D-HEVC, the base view, multiple dependent views and depth maps, are required to be coded as a single stream, the computational complexity is high. Thus, it is desirable to design an efficient method that reduces this complexity, without, of course, hampering 3D-HEVC's coding efficiency.

1.1.3.2.3 Overview of Complexity Reduction Techniques Designed for 3D-HEVC

In order to reduce to complexity of the 3D-HEVC encoder, several complexity reduction methods have been proposed [73]–[85]. In [73], based on a synthesized view difference model a low complexity adaptive view synthesis optimization has been proposed for the 3D-HEVC. In another study, researchers propose a low complexity neighboring block based disparity vector derivation [74]. In [75], researchers modify the merge candidate list to reduce the cost of the disparity compensation prediction. In addition, this method uses the classification results to enable and disable the disparity search. In [76], a low complexity mode decision method is proposed. This method classifies each tree block into near, middle, and far regions. Then, based

on this classification, appropriate modes are suggested. In addition, this method uses the classification results to enable and disable the disparity search. In [77], researchers propose an early skip detection method, which uses the spatial and interview correlations in 3D-HEVC. In [78], it is suggested to use the information of the previously encoded view frames and neighbouring blocks in the current view to predict the motion search range and skip motion estimation (ME) and disparity estimation (DE). In [79], [80], a fast encoder decision method for coding views is proposed that uses the five inter-view neighbouring blocks in the reference view to detect the Merge mode and to terminate the quad-tree partitioning. In [81], a fast mode decision algorithm is proposed for 3D-HEVC that uses the mode correlation between depth levels, the correlation among the views and depth maps, and correlation among the spatial-temporal blocks to predict the skip mode and mode candidates for the to-be-coded blocks. Researchers in [82], suggest to exclude the additional 3D-HEVC merge candidates from the candidate list for the prediction units of size 8x4 and 4x8. In another studies, several complexity reduction methods are proposed for depth map coding in 3D-HEVC [83], [84]. In order to reduce the complexity of simplified depth coding (SDC), in [83] it is suggested to perform intra prediction at a subsampled level. In [84], a gradient-based complexity reduction algorithm is proposed for depth-maps Intra prediction of 3D-HEVC. This method applies a gradient based filter in the encoding block borders to anticipate the best positions for evaluating Wedgelets. Although the methods mentioned above reduce the computational complexity of different procedures of 3D-HEVC, they do not deal with the computational complexity of the exhaustive mode search and motion estimation, two of the most complex processes of 3D-HEVC encoding.

1.2 Thesis Contributions

In this thesis, we present novel methods designed to reduce the computational complexity of the SHVC and 3D-HEVC standards. Our methods are implemented on MPEG's SHVC reference software model (SHM) and 3D-HEVC reference software test model (3D-HTM). As such, our contributions are ready to be used by the industry. Note that all our methods are introduced at the encoder side with the decoder left untouched. The performance of our methods is compared with that of the unmodified reference codecs in terms of execution time and compression performance, as recommended by MPEG. Related contributions were submitted and presented at the ITU/MPEG meetings.

In Chapter 2, first an adaptive search range adjustment method is proposed for spatial scalability, which reduces the number of search points in motion estimation. In Section 2.2, a probabilistic method is presented for quad-tree partitioning of spatial SHVC, which predicts the CU size and thus reduces the complexity of the quad-tree. Finally, we use the combination of these two methods to reduce the overall complexity of spatial scalability. Unlike the existing methods, which can only reduce the complexity of intra prediction, the combination of our methods reduces the computational complexity of inter prediction as well as the intra prediction. Even in the case of only intra prediction, our probabilistic based quad-tree partitioning method outperforms the existing methods. Experimental results show that the combined method outperforms every other existing method and reduces the total computational complexity of SHVC by more than 39.25%.

In Chapter 3, we present four complexity reduction schemes for the enhancement layer of the quality extension of HEVC. In Section 3.1, a content adaptive complexity reduction scheme is proposed which reduces the mode search by introducing an early termination approach as well

as the motion estimation number of search points for quality SHVC. Results show that our content adaptive complexity reduction method reduces the total encoding time about 29.93% on average at the cost of 1.22% bitrate increase for EL1 and 2.37% bitrate increase for EL2. In order to further reduce the complexity of the search mode, in Section 3.2, we design three mode prediction schemes for quality scalability. First, in Section 3.2.1, using statistical analysis we propose a hybrid complexity reduction scheme for mode prediction. Performance evaluations show that our hybrid complexity reduction scheme reduces the total encoding time about 48.49% at the cost of 2.10% bitrate increase for EL1 and 3.05% bitrate increase for EL2, on average. Our study also indicates that the correlation between the content and the coding configuration of the training and the test videos determines the efficiency of the hybrid complexity reduction scheme. In order to reduce this dependency, in Section 3.2.2, we present a Bayesian based mode prediction method for quality SHVC, which uses a set of probabilistic models and the encoding information of the early frames of each scene to reduce prediction error. This method achieves the total encoding time reduction of 36.84% at the cost 1.63% bitrate increase for EL1 and 2.12% bitrate increase for EL2, on average. In order to further improve the efficiency of our Bayesian based method, in Section 3.2.3, we propose an online-learning based inter/intra mode prediction method which is an extension of the Bayesian method. This method gradually fine-tunes its model during the course of encoding. Unlike the methods proposed by other researchers for SHVC's complexity reduction [61]–[63], the methods proposed in Sections 3.2.2 and 3.2.3 use a set of probabilistic models for mode prediction that is updated for each new video, making them adaptively fine-tuned for new content and different coding configurations. Our results show that our online-learning based mode prediction method reduces the computational complexity of SHVC's encoder about 45.40% at the cost of 0.67% bitrate increase for EL1 and 1.13% bitrate

increase for EL2, a significant improvement over our previous methods and the state-of-the-art complexity reduction scheme (the early merge mode detection method) proposed in [28]. Finally, we combine our best mode prediction approach (online-learning based mode prediction method) with the content adaptive complexity reduction scheme. The combined method reduces the total encoding time by about 51.60% at the cost of 0.86% bitrate increase for EL1 and 1.40% bitrate increase for EL2, on average.

In Chapter 4, an adaptive search range adjustment and early termination method for exhaustive mode search are proposed for dependent texture views of 3D-HEVC (Section 4.1). In addition, a Bayesian based mode prediction is proposed for the exhaustive mode search process of 3D-HEVC (Section 4.2). Unlike the 3D-HEVC complexity reduction methods proposed by other researchers, the method presented in Section 4.2 use a probabilistic model for mode prediction that is created using training data. Then, this probabilistic model is updated using the coding information of early frames of each new video, improving the mode prediction accuracy. Our experiments show that our proposed methods (Section 4.1 and Section 4.2) reduce the execution time of 3D-HEVC's encoder significantly. Finally, we propose an online-learning based complexity reduction method in Section 4.3 that incorporates the two above-mentioned methods (Section 4.1 and Section 4.2) to achieve the highest complexity reduction. Using the online learning approach improves the prediction accuracy and augments the achieved complexity reduction performance. For dependent texture views, our online learning approach reduces encoding time by 67.70% on average.

2. Complexity Reduction Methods for Spatial Scalability

As already discussed in Chapter 1, SHVC uses all the advanced coding tools of HEVC and the coding tools specially designed for scalable video coding to encode several ELs in addition to the BL. One major barrier in the wide-spread adaption of SHVC standard is its computational complexity.

The existing complexity reduction method proposed for spatial SHVC by other researchers is only able to reduce the computational complexity of all intra coding [63]. Thus, it does not deal with the complexity of motion estimation process of inter prediction, one the most complex procedures of SHVC. To address this issue, in this Chapter, we propose an adaptive search range adjustment, which reduces the computational complexity of inter prediction and thus the complexity of the motion estimation process of spatial SHVC. This method is designed based on statistical studies. The other main factor that contributes to the SHVC encoder complexity is choosing the best partitioning structure for the coding tree units (CTUs). In this regard, we propose a quad-tree partitioning method at the CU level. The proposed method uses the CTU partitioning structure of the already encoded CTUs in the enhancement layers (ELs) and base layer (BL) to predict the coding unit sizes of the to-be-encoded CTUs in the EL. The method creates its probabilistic model using the training videos and it fine-tunes its model for each new video using the early frames of the scene. Then, it uses Bayesian classifier to predict the partitioning structure. Performance evaluations confirm that the combination of our proposed complexity reduction schemes significantly reduces the execution time of the SHVC encoder, while maintaining the overall quality of the coded streams.

The rest of the Chapter is divided as follows. Section 2.1 presents our adaptive search range adjustment for spatial scalability. In Section 2.2, our quad-tree prediction method is presented.

Section 2.3 shows our experimental results and analysis. Finally, conclusions are drawn in Section 2.4.

2.1 Adaptive Search Range Adjustment

In the inter-prediction process, selecting a large search range leads to high computational costs, while selecting a small search range produces poor matching results. The optimal motion search range would result in reduced complexity without hampering the compression performance.

The main goal of our adaptive motion search method is to decrease the computational cost of the explicit EL motion vector search by reducing the number of examined blocks with a minimal effect on compression performance. Two types of motion search algorithms are supported by SHVC, namely the full search and fast search. The full-search algorithm examines all the blocks within a search-window to find the best match. The fast motion search algorithm examines only the blocks that are more likely to generate sub-optimal motion vectors [86]. In fast motion, several inspection points at different distances (power of two) from the center of the search-window are used for motion estimation [86]. Similar to the idea of fast search, our scheme tries to restrict the number of inspection points in EL using the BL motion information. To this end, we first studied the relationship between the BL and EL motion vectors. A representative set of video sequences including the four different videos (BQMall, Racehorse , PartyScene, and Vidyo3) are encoded using the SHVC reference software (SHM. 3.0) with Random Access High Efficiency (RA-HE) configuration (Hierarchical B pictures, Group of Pictures (GOP) size of 8, Sample adaptive offset (SAO), and Rate distortion optimized quantization (RDOQ) are enabled). In our experiments, we use one EL in addition to the BL. The quantization parameters (QPs)

Table 2.1 Motion vector distribution of the EL, given the distribution of motion vector of the BL.

BL's SR	Appropriate SR for EL					
	|MVs|≤SR/32	|MVs|≤2*SR/32	|MVs|≤2*SR/16	|MVs|≤2*SR/8	|MVs|≤2*SR/4	|MVs|>2*SR/4
|MVs|≤SR/32	19.11%	90.60%	94.38%	98.90%	99.25%	0.75%
|MVs|≤SR/16	7.00%	16.92%	94.75%	97.72%	99.23%	0.77%
|MVs|≤SR/8	5.51%	15.85%	21.79%	92.33%	97.34%	2.66%
|MVs|≤SR/4	5.21%	17.61%	19.25%	25.20%	94.16%	5.84%
|MVs|≤SR/2	5.10%	19.39%	20.24%	22.29%	26.86%	73.14%
|MVs|>SR/2	4.19%	17.11%	34.90%	37.21%	42.10%	57.90%

used for the base layer and enhancement layer (QPB, QPE) are as follows: (22, 22), (26, 26), (30, 30) and (34, 34). The diamond fast search is used with search range (SR) of 64. In this case, only corner points of the diamonds, which are in six different distances from the center of the search-window are tested. The inspection intervals include $SR/2^5$, $SR/2^4$, $SR/2^3$, $SR/2^2$, SR/2, and SR [86].

In the case of spatial scalability, the MVs of the base layer and those of the enhancement layer are highly correlated. The relationship between the largest MVs of the CTUs in the BL and those of the co-located CTUs in the EL is reported in Table 2.1 for spatial ratio of 2. Note that the statistical information reported in Table 2.1 is calculated by averaging the results obtained from different QP settings for each video sequence. We observe that the chances the largest size MV of each EL CTU falls into the interval two times larger than the size MV of the corresponding BL CTUs are more than 73.14% (for the case that largest size MV falls into an interval smaller or equal to SR/2). Based on the above observation, we propose an adaptive search range adjustment method for the spatial scalability. The method first classifies the CTUs within each frame in the base layer to different classes based on their motion information and adjusts the search range of the co-located CTUs in the enhancement layer accordingly as follows [87]:

$$SR_{CTU} = \begin{cases} round\left(\alpha \frac{SR}{32}\right) & if\ more\ than\ 85\%\ of\ MVs\ have\ \ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{32} \\ round\left(\alpha \frac{SR}{16}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ \ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{16} \\ round\left(\alpha \frac{SR}{8}\right) & else\ \ if\ more\ than\ 85\%\ of\ MVs\ have\ \ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{8} \\ round\left(\alpha \frac{SR}{4}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{4} \\ round\left(\alpha \frac{SR}{2}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ \ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{2} \\ SR & Otherwise \end{cases} \tag{2.1}$$

where SR_{CTU} is the adjusted search range of the CTUs in the enhancement layer (the largest coding block in SHVC is called CTU), SR indicates the search range defined in the encoder settings (configuration file), MVs represent the motion vectors of the co-located CTU in the base layer, and α indicates the spatial scalable ratio between the width/height of the EL and that of BL. Note that since the resolution of EL is α times larger than the resolution of BL, to find the co-located LCU in BL, the coordinates of CTU in EL should be divided by α (see Figure 2.1). As (2.1) shows, the search range of the CTU in the enhancement layer is adjusted based on the amplitude of the MVs of the co-located CTU in the base layer. Note that all the CUs within the CTU in the enhancement layer will have the same adjusted motion search range setting. As it can be observed from (2.1), the search range of the CTUs in the enhancement layer can become quite small, depending on the motion information of the co-located CTUs in the base layer,

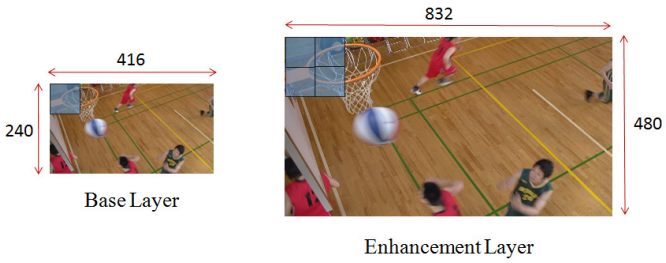

Figure 2.1 Example of the CTU in BL and four corresponding CTUs in EL, when the width and height of the EL are two times larger than those of BL.

which can significantly reduce the overall computational cost.

2.2 Probabilistic Approach for Predicting the Size of Coding Units in the Quad-tree Structure of the Spatial Scalable HEVC

As mentioned before, SHVC encoder checks several tree structures to find the best CTU partitioning structure for each CTU. Therefore, the process of determining the CTU structure is one of the most time consuming operations of the encoder. One way to reduce this complexity is to predict the CU sizes in the quad-tree structure of the EL CTUs so that SHVC encoder does not need to check all the possible CU sizes in the quad-tree structure. One way of achieving this is to use a machine learning approach that generates a model that efficiently predicts the CTU structure of the to-be-encoded EL CTU. To achieve this goal the information of already encoded CTUs can be utilized. In the case of SHVC, since there is a high correlation between the base layer and the enhancement layers, we propose to use the CTU structure of the four neighboring blocks of the current to-be-encoded CTU in the EL, its corresponding CTU (or CTUs) in the reference layer (or layers), and its corresponding CTU in the previous frame in the EL as predictors for the CTU structure prediction.

2.2.1 Model Generation

To generate a model, we propose to use a supervised learning. A supervised learning consists of a training and test procedures. One of the main steps of generating our model is to find an appropriate labeling system to translate the tree structures of the CTUs into labels (numbers) so that they can be used in the training and test procedures. Considering the maximum depth partitioning is set into 4 and the CTU size is equal to 64×64 in most of the SHVC configurations,

we investigate different labeling systems. The first option for labeling system is using the actual CTU structures (at CU level) as the labels [88]. More precisely, we can find all the possible CTU structures and assign different numbers or symbols to each. As illustrated in Fig. 2.2.a, each CTU of size 64×64 can be split into four blocks of size 32×32. Each 32×32 block then can be kept as it is (Fig.2.2.b-2) or it can be split into smaller blocks in 16 possible ways (see Fig.1-b-3 to Fig.1-b-18). Therefore, there are 17 partitioning structures (see Fig.2.2.b-2 to Fig.2.2.b-18) for each 32×32 block. Since the encoder can split each CTU into four blocks of size 32×32, which can also be split to smaller blocks (17 possible options), in total there are 17^4+1 (=83522) possible partitioning structures for each CTU. We call this labeling system "CTU based labeling system". In this system, we need a large training video dataset to cover all the possible CTU structures given huge number of possible combinations for the predictor-CTUs structure. In this case, we also need large memory capacity to store the model. To address these issues, we propose to partition each frame into imaginary sub-CTUs of size 32×32 (see Fig.2.2.a). Then assign a label (number) to each sub-CTU. If the sub-CTU does not need to be split, we assign it with number 2 as its label (Fig.2.2.b label 2). If a sub-CTU is merged with other sub-CTUs and construct a CU of size 64×64, we assign number 1 as its label (Fig. 2.1.b label 1). Other than these two cases there are 16 possible structures to partition a sub-CTU to CUs of different sizes as illustrated in Fig 2.1.b (label 3 to 18). As it is observed in this labeling system the number of possible sub-CTU structures is reduced to 18 possible cases. This labeling system is called "sub-CTU based labeling system" in our study. Considering the total number of labels is reduced into 18, a small training video dataset is required to cover all possible structures given the possible sub-CTU structures of the predictors. In this case even few number of frames can be used for training and still cover all the cases. In addition, this labeling system makes the training process and the

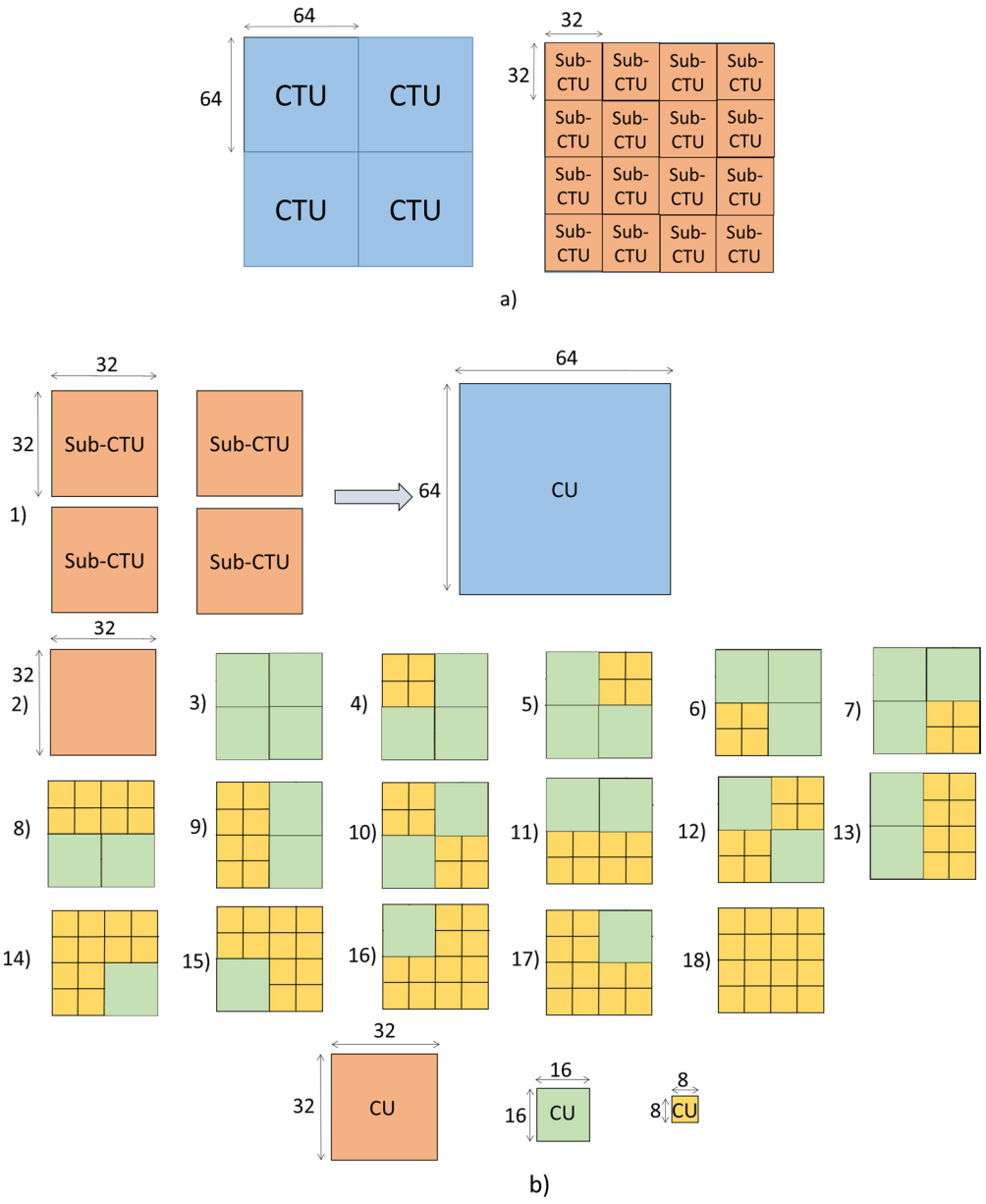

Figure 2.2 a) An example of a frame with four CTUs that each CTU is split into four imaginary sub-CTUs. b) The labels (numbers) that show the 18 possible structures for sub-CTU partitioning and merging.

classification process faster compared to case of using CTU based labeling system. Moreover, it reduces the memory usage significantly. Note that the sub-CTU based labeling system can be modified for the CTU of different sizes (for the cases that the CTU sizes are different from 64×64). In this study, we call the number, which is assigned for each of the possible sub-CTU partitioning structures (at the CU level) as a label [88].

These labeling systems can be used for spatial scalability. In our implementation, for the spatial scalability we assume there is one EL in addition to the BL.

Fig. 2.3 illustrates an example of a sub-CTU in the EL, whose structure needs to be predicted using the structure (label) of its six predictor sub-CTUs (i.e., corresponding base layer sub-CTU, its corresponding sub-CTU in the previous frame (T_0=t-1), and its four neighbors in the current enhancement layer).

The objective here is to predict the CU sizes in the CTU structure, so that the encoder is not required to check all the CTU structures for the EL CTUs and, thus, significantly reducing the computational complexity. This can be modeled as a supervised learning problem that is resolved through two stages of training and testing. During the training process, the encoder encodes the BL using the unmodified SHVC [58]. Each frame is split into several CTUs of the same size, which in turn consist of one or more CUs. SHVC checks all the available inter/intra prediction modes, calculates the RD cost of each mode and, at the end of the process, the CTU structure that leads to the lowest RD cost is chosen. Each EL is also encoded using the conventional SHVC. As in BL, all the available CTU structures are checked to find the CTU structure with the

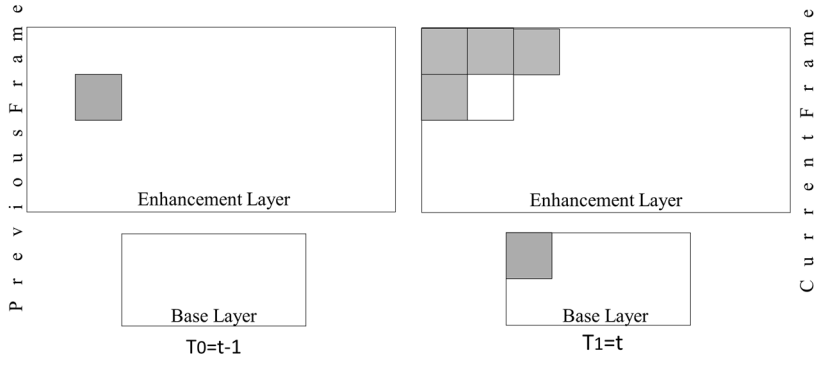

Figure 2.3 Current EL sub-CTU (white block) and its six predictors (Gray blocks) when the spatial scalable ratio is 2 for the spatial scalability.

lowest rate distortion cost. For each CTU in the BL and the EL(s), the information about the sub-CTU labels in the BL and EL(s) is used to update the probability of each sub-CTU structure in the current sub-CTU in EL. These conditional probabilities are later used in the testing process.

In rest of this Section, we present the mathematical foundation of our modeling. As mentioned before, a probability can be assigned to each label of the current sub-CTU, given the labels of its predictor sub-CTUs (i.e., *P(current sub-CTU structure| predictor sub-CTU structures)*). Assume Y is the random variable corresponding to the label of the current sub-CTU at current EL, and X is a random vector corresponding to the label of its predictor sub-CTUs. The posterior probability $P_l(Y|X)$ of the to-be- encoded sub-CTU at the enhancement layer, l given the labels of the predictor sub-CTUs, can be calculated using the following Bayes rule:

$$P_l(Y|X) = \frac{P(X|Y)P_l(Y)}{P(X)}, \qquad l=1,..,L \tag{2.2}$$

In order to train our classifier, we need to determine each of the $P_l(X|Y)$ and $P_l(Y)$. $P_l(Y)$ is our prior of the label of the to-be encoded sub-CTU at the enhancement layer l (i.e., $l=1$ for the EL1 and $l=2$ for the EL2). L shows the number of enhancement layers used. In this study, $L=1$ for spatial scalability. $P_l(X|Y)$ is the class-conditional density, which defines the distribution of data that is expected to be seen in each class. Here, $P_l(X|Y)$ indicates the probability of the labels of the predictor sub-CTUs, given each label of the current sub-CTU at the l^{th} EL (i.e., *P(predictor sub-CTU structures| structure of current sub-CTU at* l^{th} *EL)*). Suppose that there are M ($M=18$ in our case) different possible partitioning structures (labels) for different sub-CTUs, which would result in M different values for the random variable Y. Also, X is a vector with N components, each taking M possible discrete values that result in M^N-1 different values for the random vector X. In this study, N is equal to the number of predictors. Since the probability should sum to one, the learning algorithm needs to estimate $M-1$ different parameters to estimate

$P_l(Y)$. However, estimating $P_l(X|Y)$ requires learning an exponential number of parameters, i.e., $M(M^N-1)$, which is an intractable problem. Thereby, the key to use the Bayes rule is to specify a suitable model for $P_l(X|Y)$.

In order to overcome the above-mentioned intractability problem, we used the Naive Bayes classifier [89]. The Naive Bayes classifier dramatically reduces the complexity of estimating $P_l(X|Y)$ by making a conditional independence assumption [90]. This learning algorithm assumes that, given Y, the different members of the X vector are independent. Considering this conditional independence assumption, $P_l(X|Y)$ is computed as follows:

$$P_l(X|Y) = P_l(X_1, X_2, \dots, X_N|Y) = \prod_{n=1}^{N} P_l(X_n|Y) \tag{2.3}$$

Therefore,

$$P_l(Y|X) = \frac{\prod_{n=1}^{N} P_l(X_n|Y)\, P_l(Y)}{P(X)} \qquad , l=1,..,L \tag{2.4}$$

Thus, this simplifying assumption makes the representation of $P_l(X|Y)$ simpler and reduces the number of parameters from an exponential term to just M^2N.

According to the optimal Bayes decision rule [91], the label of the posterior probability distribution is the predicted label of the current sub-CTU. Therefore, for classifying a new X, we use the following formula:

$$\mathrm{y_m} = \underset{y_m}{\operatorname{argmax}} \frac{P_l(Y=y_m) \prod_{n=1}^{N} P_l(X_n|Y=y_m)}{\sum_{m=1}^{M} P_l(Y=y_m) \prod_{n=1}^{N} P_l(X_n|Y=y_m)} = \underset{y_m}{\operatorname{argmax}}\, P_l(Y=y_m) \prod_{n=1}^{N} P_l(X_n|Y=y_m), \tag{2.5}$$

where y_m is the m^{th} possible value of Y. The normalization part, $P_l(X)$, of the posterior distribution has been omitted due to the fact that the denominator does not depend on y_m. The resulting y_m is the predicted sub-CTU structure for the current to be encoded CTU.

To find the optimal value of y_m in equation 2.5, we need to know $P_l(X|Y)$ and $P_l(Y)$. These probabilities are computed during the training process. One well known approach for estimating these probabilities is using the Maximum Likelihood Estimation (MLE) [89]. The MLE estimate for $P_l(Y)$ is:

$$P_l(Y = y_k) = \frac{N_k}{Total\ number\ of\ tries}, \tag{2.6}$$

where the element N_k denotes the number of observed instances of class y_k. That is, N_k indicates the number of times the labels of the current sub-CTU is equal to y_k. On the other hand, the MLE estimate for $P_l(X|Y)$ is:

$$P_l(X_n = x_{nm}|Y = y_k) = \frac{N_{nmk}}{Total\ number\ of\ tries}, \tag{2.7}$$

where N_{nmk} is the number of times $X_n=x_{nm}$ has been observed in the instances of class y_k [91]. That is, N_{nmk} denotes the number of times the labels of the n^{th} predictor is equal to x_{nm}, while the label of the current sub-CTU is equal to y_k.

A major problem arising from using MLE to estimate probabilities is when $P_l(X=x_i|Y)=0$, which makes the Bayesian equation (2.3) equal to zero. This means that we have not considered some possible sub-CTU structures in the training set. This is an example of over-fitting [41]. We address this issue by employing the Maximum a Posteriori (MAP) [89] estimation. MAP estimate is a regularization of MLE that resolves the above-mentioned over fitting problem by incorporating a prior distribution over the parameter that we want to approximate. Since our objective is to predict the CU sizes of the current EL, a prior distribution will be assigned using the prior knowledge of sub-CTU structures.

Here, the distribution of the $P_l(Y|X)$ is a categorical distribution, due to the fact that the number of possible sub-CTU structures (labels) is more than two. Thus, the MAP estimate for $P_l(Y)$ is:

$$P_l(Y = y_k) = \frac{N_k + \alpha_k}{Total\ number\ of\ tries + \sum_{k=1}^{M} \alpha_k}, \quad (2.8)$$

where α_k determines the strength of prior assumptions relative to the observed data and M is the number of different sub-CTU structures/values that Y can take.

The MAP estimate for $P_l(X|Y)$ is:

$$P_l(X_n = x_{nm}|Y = y_k) = \frac{N_{nmk} + \alpha_{nmk}}{Total\ number\ of\ tries + \sum_{m=1}^{M} \alpha_{nmk}}, \quad (2.9)$$

where α_{nmk} determines the strength of prior assumptions relative to the observed data and M is the number of distinct values which X_l can take. Note that we will call α_k and α_{nmk} hyper parameters (initial model), hereafter.

2.2.1.1 Finding Corresponding Sub-CTU in the Reference Layer

For spatial scalability, the corresponding sub-CTU in the BL is determined based on the spatial ratio between the resolution of EL and BL (see Fig. 2.3). Suppose that (x_{EL}, y_{EL}) shows the location of the top left corner of a sub-CTU in EL. Then, the location of the top left corner of the corresponding sub-CTU in the BL is:

$$(x_{BL}, y_{BL}) = \left(\left[\frac{x_{EL}}{\frac{W}{2}r} \right] \frac{W}{2}, \left[\frac{y_{EL}}{\frac{W}{2}r} \right] \frac{W}{2} \right), \quad (2.10)$$

where r is the spatial ratio and [x] denotes the largest integer not greater than x. W denotes the width of the CTU that is set in the configuration (in this study W=64).

2.2.2 Model Training and Testing

The first step for supervised learning is training. In this step, we need to convert the CTU-structures into numbers. If the CTU based labeling system is utilized, the sizes of the $P_l(X|Y)$ and $P_l(Y)$ tables are 83522×83522 and 83522×1, respectively. Since the number of learning parameters in the probability tables is very large, we have an over-fitting problem. We avoid this

problem by using our sub-CTU based labeling system, reducing the size of the $P_l(X|Y)$ and $P_l(Y)$ tables to 18×18 and 18×1, respectively. This means that we need only a small training dataset to initialize our model. Our training video set is four representative sequences selected from the MPEG dataset [92] for the HEVC call for proposals video sequences. These are: PartyScene (832×480, 50fps), BQMall (832×480, 60fps), Racehorse (832×480, 30fps), and Vidyo3 (1280×720, 60 fps). In order to find the priors, we encode our training video sequences and store their CTU information. Equations (2.6) and (2.7) are utilized to compute the priors (hyper parameters). Since in the case of using sub-CTU based labeling system we can cover all the possibilities for $P_l(X|Y)$ and $P_l(Y)$, we use the first second (e.g., 50 frames if the frame rate is 50fps) of the to-be-coded (test) video to fine-tune our model for the test video. We call these frames the training frames, hereafter. More precisely, the first second of a scene is coded using the unmodified SHVC, then the sub-CTU structure information of all the encoded frames is utilized to fine-tune our model using (2.8) and (2.9). The fine-tuning process adjusts the model based on the video content, the quantization parameters, and configuration.

Note that in the testing process – similar to the training process - we use the sub-CTU based labeling system. This makes the testing process much faster than if the CTU based labeling system was used. This is because the number of multiplications (Eq. (2.4)) required for computing $P_l(Y|X)$ is significantly reduced compared to case of using CTU based labeling system. In addition, in this case, we predict the probability of only 18 sub-CTU structures and find the most probable sub-CTU structures among only 18 structures. Therefore, the sorting process (Eq. (2.5)) results in a negligible time-execution delay. In order to predict a sub-CTU structure, first we use the predictor sub-CTUs that are available to predict sub-CTU structures in the current EL CTU. In the case that none of the neighboring sub-CTUs is available, for

prediction we use the corresponding sub-CTU in BL, the corresponding sub-CTU in the previous EL if available, and the corresponding sub-CTU in the previous frame. In order to predict the sub-CTUs more accurately, the probability of the suggested sub-CTU structure should be very high. Therefore, we consider 0.95 as the probability threshold that the most probable sub-CTU structures should reach in order to be considered as the reliable predicted structure candidates. We use our Bayesian method with all the available predictor sub-CTUs for predicting the current EL sub-CTU. First, the four most probable sub-CTU structure is found and, if the probability of the first most probable sub-CTU structure is more than 0.95, then this sub-CTU structure is suggested as the candidate sub-CTU structure. Otherwise, if the accumulative probability of the first and second sub-CTU structures is more than 0.95, then both are suggested as the candidate sub-CTU structures. If the first and second candidates cannot fulfill the probability condition, we check the third and fourth structure. If they all fail, our scheme checks all the possible partitioning structures for the to-be-encoded CTU. If sub-CTU structure candidates have been identified for all four sub-CTUs, then the structure of that CTU may be determined.

At this point, CU size refinement may be needed, if the suggested sub-CTU predictions do not comply with standard CTU partitioning-rules. This may happen in the case that for at least one of the sub-CTUs, the CU size of 64×64 is suggested, and, for at least one of the other sub-CTUs in the same CTU, a different CU size is recommended. In this case, for each sub-CTUs that the 64×64 size CU is suggested, the next most probable sub-CTU structure is considered as a candidate. Afterwards, the new suggested sub-CTU structures are utilized to make the CTU structure candidate. Then, the 64×64 size CU will also be checked.

Note that during sub-CTU structure prediction, for the case that the reference predictor sub-CTU and the current sub-CTU belong to the same CTU, the most probable label of the reference

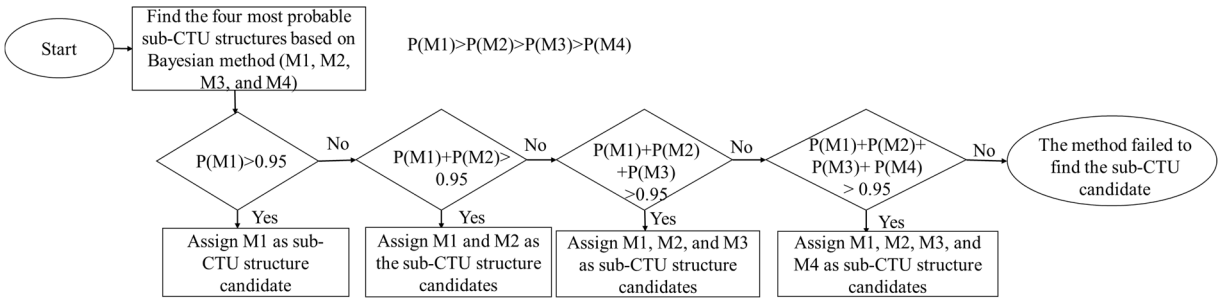

Figure 2.4 Diagram of the proposed sub-CTU structure prediction method.

predictor sub-CTU is utilized to predict the current sub-CTU structure. In this case, if for the reference predictor sub-CTU, more than one sub-CTU structure is suggested, our method first checks the candidate sub-CTU structures for that predictor sub-CTU. Then, based on the encoder decision for that predictor sub-CTU, we modify the sub-CTU prediction for the current sub-CTU. The block diagram of the proposed CTU prediction structure is shown in Fig. 2.4. We call the combination of the method presented and the refinement process the CTU prediction method.

Note that the unmodified SHVC will check all the possible CTU structures. For all the CUs, the inter/intra prediction exhaustive mode search process is used. However, our proposed method checks only a few CTU partitioning structures (at CU levels) for the EL, therefore, reducing the complexity of the SHVC significantly.

2.3 Experimental Results and Discussions

We use eleven sequences to evaluate the performance of the proposed complexity reduction schemes. Table 2.2 summarizes the specifications of this test dataset [92], [93]. All the methods used are implemented in the SHVC reference software (SHM 6.1 [94]). In our implementation, we have one EL in addition to the BL for the spatial scalability. As suggested in common SHM test conditions [93], we test our proposed method for two spatial ratios (x1.5 and x2) [93]. The

Table 2.2 Test video dataset specifications.

Name	Resolution	Frame Rate (fps)
Traffic	2560×1600	30 Hz
PeopleOnStreet	2560×1600	30 Hz
Kimono	1920×1080	24 Hz
ParkScene	1920×1080	24 Hz
Cactus	1920×1080	50 Hz
BasketballDrive	1920×1080	50 Hz
BQTerrace	1920×1080	60 Hz
BasketballDrill	832×480	50 Hz
BlowingBubbles	416×240	50 Hz
BasketballPass	416×240	50 Hz
BQSquare	416×240	60 Hz

BL layer is generated by down-sampling the original video streams using the DownConvert program in SHM 6.1.

The performance of our proposed complexity reduction methods is compared with that of the unmodified SHVC encoder in terms of execution time, and impact on bitrate and PSNR. For the spatial scalability, we have examined four scenarios for the EL: 1) Early Merge Decision [28] (just for the EL), 2) Adaptive Search Range Method (ASRM) for Spatial Scalable HEVC, 3) our proposed CTU prediction method (PCPM), and 4) The combination of ASRM and PCPM.

In our comparative study, we use the random access (RA) main [95], [96] and intra main [97] configurations of SHVC. SAO and RDOQ are enabled [98]. The quantization parameters of the BL and EL were set to (QPB, QPEL) = {(22, 22), (26, 26), (30, 30), (34, 34)} [93]. As mentioned before, the ASRM reduces the complexity of motion estimation for the inter prediction. The EMD method reduces the complexity of mode prediction step of the encoder. On the other hand, our PCPM reduces the complexity of CTU partitioning of SHVC. Among all the methods (ASRM, EMD, and PCPM), only PCPM is able to reduce the complexity of the SHVC encoder under the intra main configuration, we report only the results of PCPM for the intra main configuration.

Table 2.3 The impact of all the methods on bitrate and PSNR for the spatial scalability under RA main configuration.

Video Sequence	Scalable Ratio	Methods							
		ASRM		EMD		PCPM		PCPM+ASRM	
		BD-PSNR (dB)	BD-BR	BD-PSNR (dB)	BD-BR	BD-PSNR (dB)	BD-BR	BD-PSNR (dB)	BD-BR
Traffic	2	-0.005	1.14%	-0.010	3.30%	-0.087	2.85%	-0.089	3.12%
PeopleOnStreet	2	-0.023	0.63%	-0.057	1.30%	-0.072	1.60%	-0.079	1.76%
Kimono	1.5	-0.018	0.10%	-0.026	0.90%	-0.025	0.89%	-0.031	0.92%
	2	-0.028	0.74%	-0.061	2.08%	-0.047	1.56%	-0.058	1.83%
ParkScene	1.5	-0.034	0.31%	-0.090	2.99%	-0.034	1.13%	-0.044	1.21%
	2	-0.051	0.83%	-0.135	4.23%	-0.065	2.09%	-0.084	2.36%
Cactus	1.5	-0.009	0.15%	-0.024	1.44%	-0.025	1.41%	-0.027	1.44%
	2	-0.015	0.45%	-0.048	2.52%	-0.039	2.05%	-0.044	2.19%
BasketballDrive	1.5	-0.012	0.28%	-0.047	2.47%	-0.031	1.68%	-0.035	1.77%
	2	-0.017	0.82%	-0.053	2.66%	-0.044	1.70%	-0.052	2.07%
BQTerrace	1.5	-0.007	0.25%	-0.059	3.34%	-0.024	1.87%	-0.026	1.94%
	2	-0.010	0.90%	-0.073	4.26%	-0.039	2.68%	-0.042	2.97%
BasketballDrill	2	-0.011	0.75%	-0.084	2.06%	-0.080	1.88%	-0.085	2.16%
BlowingBubbles	2	-0.033	0.95%	-0.026	1.32%	-0.032	1.22%	-0.046	1.59%
BasketballPass	2	-0.150	1.25%	0.113	2.27%	-0.089	1.72%	-0.124	2.20%
BQSquare	2	-0.110	1.01%	-0.220	5.94%	-0.059	2.90%	-0.094	3.28%
Average	1.5	-0.016	0.22%	-0.049	2.23%	-0.028	1.39%	-0.033	1.45%
	2	-0.041	0.86%	-0.063	2.90%	-0.059	2.02%	-0.073	2.32%

The results of our experiment are reported in Table 2.3, Table 2.4, Table 2.5, and Table 2.6. For each method, we report the time reduction percentage compared to the unmodified SHVC encoder, the impact on the bitrate (Bjontegaard delta rate (BD-BR) [99]), and the video quality in terms of PSNR (BD-PSNR [99] in dB). For spatial scalability, two time reduction percentage values are reported, the time reduction percentage of the EL and total time reduction percentage (BL+EL). In our experiment we used a Blade with an Intel Xeon X5650 6-core processor, running at 2.66GHz, and 8-GB RAM from the Bugaboo Dell Xeon cluster from WestGrid (a high performance computing consortium in Western Canada). In Tables III and V we report the CU size prediction accuracy (Acc%) of our CTU prediction method. The Acc% was computed using the following formula:

$$Acc\% = \sum_{f=1}^{F} \sum_{j=1}^{J} \frac{\frac{\sum_i s_{CU_{ijf}} \times \omega_{CU_{ijf}}}{\sum_i s_{CU_{ijf}}}}{J \times F} \times 100\%, \quad (2.11)$$

where CU_{ijf} indicates the i^{th} CU of the j^{th} CTU of the f^{th} frame of the test video. $S_{CU_{ijf}}$ represents the area (number of pixels in width × number of pixels in height) of the CU_{ijf}. $\omega_{CU_{ijf}}$ is equal to one if the CU size of CU_{ijf} is predicted correctly (otherwise $\omega_{CU_{ijf}} = 0$). J and F are the total number of CTUs in each frame and total number of frames of the test video, respectively.

Table 2.3 shows the BD-PSNR and BD-BR results for the spatial scalability for the random access configuration. For the spatial scalability of scalable ratio 1.5, we observe that if the EMD is used, the average BD-BR value is about 2.23% for EL (see Table 2.3). For this method, the average EL execution time and average total (BL+EL) execution time reductions are 57.97% and 37.40%, respectively. ASRM achieves an average of 24.61% time reduction for the EL and 14.95% time reduction for BL+EL compared to the unmodified SHVC. The average BD-BR value for ASRM is 0.22%. Our proposed PCPM method reduces the EL execution time by an average of 67.87% at the cost of 1.39% bit-rate increase compared to the unmodified SHVC encoder (see Tables 2.3 and 2.4). Our method decreases the total execution time for BL+EL by 44.45% on average. The time reduction achieved by our proposed CTU prediction method for the EL and BL+EL are on average 9.91% and 7.05% compared to EMD (see Table 2.4), respectively. Compared to the ASRM, the time reduction performance of the CTU prediction method is 43.26% and 29.51% for EL and BL+EL, respectively. We can also observe from Tables 2.3 and 2.4 that PCPM+ASRM achieves EL execution time reduction of 74.86% at a cost of 1.45% average bitrate increase for EL. For the same combination, the total (BL+EL) time reduction percentage is 51.31%, on average.

Tables 2.3 and 2.4 also show the results of the spatial ratio of 2 for the random access configuration. ASRM achieves an average of 21.92% time reduction for EL and 17.66% for BL+EL compared to the unchanged SHVC. This method leads to a 0.86% average bit-rate

increase. The EMD method reduces the coding execution time by 50.98% for EL at the cost of 2.90% increase in BD-BR. This method reduces the total execution time (BL+EL) by 41.20% compared to the unmodified SHVC. The proposed CTU prediction method decreases the EL execution time by 58.42% at a BD-BR cost of 2.02%, on average. Our PCPM achieves a total

Table 2.4 The impact of all the methods on percentage of execution time reduction (TR%) for the spatial scalability (under the random access main and intra main configurations) and the CU size prediction accuracy (Acc%) of the PCPM method.

Video Sequence	Scalable Ratio	Scalable layer	Random Access Main					All-intra Main	
			Methods					PCPM	
			ASRM	EMD	PCPM		PCPM+ASRM		
			TR%	TR%	TR%	Acc%	TR%	TR%	Acc%
Traffic	2	EL	30.56%	65.37%	67.82%	88.59%	73.44%	78.10%	94.11%
		Total	24.65%	51.67%	54.82%		59.37%	62.93%	
PeopleOnStreet	2	EL	16.97%	35.71%	55.72%	91.42%	60.21%	63.10%	94.39%
		Total	13.41%	28.23%	44.44%		47.59%	50.22%	
Kimono	1.5	EL	27.03%	62.78%	66.26%	93.01%	73.98%	70.86%	96.67%
		Total	14.32%	33.20%	35.15%		39.26%	50.25%	
	2	EL	24.54%	56.00%	58.42%	91.62%	67.24%	72.32%	94.01%
		Total	19.63%	44.89%	46.81%		57.80%	58.71%	
ParkScene	1.5	EL	29.81%	64.93%	70.36%	93.15%	78.12%	67.51%	93.22%
		Total	20.63%	45.51%	49.35%		54.80%	48.49%	
	2	EL	27.70%	62.85%	64.17%	88.41%	73.51%	65.51%	92.95%
		Total	22.31%	50.62%	51.71%		60.19%	53.39%	
Cactus	1.5	EL	22.75%	60.34%	74.80%	91.73%	79.99%	65.91%	95.22%
		Total	9.50%	36.39%	42.92%		57.93%	47.27%	
	2	EL	21.01%	56.56%	64.27%	88.94%	71.75%	71.45%	93.28%
		Total	16.68%	45.46%	51.72%		58.02%	58.27%	
BasketballDrive	1.5	EL	17.35%	41.48%	62.84%	92.14%	68.80%	68.88%	93.82%
		Total	11.49%	28.77%	47.26%		51.37%	50.25%	
	2	EL	15.85%	39.10%	44.09%	92.33%	52.90%	68.61%	89.57%
		Total	12.64%	31.12%	35.18%		42.22%	50.82%	
BQTerrace	1.5	EL	26.11%	60.31%	65.11%	90.14%	73.39%	64.88%	93.11%
		Total	18.79%	43.14%	47.59%		53.18%	47.68%	
	2	EL	23.00%	60.34%	64.20%	89.54%	71.75%	63.01%	91.42%
		Total	18.86%	49.28%	52.79%		59.02%	50.82%	
BasketballDrill	2	EL	25.75%	45.29%	57.83%	90.22%	67.58%	53.28%	90.39%
		Total	20.88%	36.73%	46.56%		54.81%	43.91%	
BlowingBubbles	2	EL	17.20%	40.33%	51.28%	92.95%	59.16%	54.48%	92.76%
		Total	14.85%	34.82%	43.98%		50.41%	45.08%	
BasketballPass	2	EL	22.28%	38.36%	54.51%	90.84%	63.48%	50.73%	90.91%
		Total	17.43%	30.32%	44.61%		51.94%	41.82%	
BQSquare	2	EL	20.12%	55.13%	59.71%	88.39%	67.28%	52.17%	96.01%
		Total	16.13%	45.61%	50.59%		57.01%	43.67%	
Average	1.5	EL	24.61%	57.97%	67.87%	92.03%	74.86%	67.61%	94.41%
		Total	14.95%	37.40%	44.45%		51.31%	48.79%	
	2	EL	21.92%	50.98%	58.42%	90.30%	66.21%	63.95%	92.94%
		Total	17.66%	41.20%	47.67%		54.40%	51.57%	

(BL+EL) execution time reduction of 47.67% compared to the unmodified SHVC. Compared to EMD, the time reductions achieved by the PCPM method for EL and BL+EL is 7.44% and 6.46%, respectively. Compared to the ASRM, PCPM reduces the execution time for the EL by 36.50% and for BL+EL by 30.01%. The PCPM+ASRM combination reduces the EL execution time by 66.21% at the cost of 2.32% bit-rate increase, on average. The same combination yields an average time reduction of 45.40% for BL+EL.

As can be seen in Table 2.4, for the random access main configuration and spatial scalability, among all the testing scenarios, the combination PCPM+ASRM for the spatial scalability outperforms every other method in terms of time reduction at the expense of a small bit-rate increase. In addition, we can also observe that the CU size prediction accuracy of our PCPM is more than 90% on average.

For the intra main configuration, as illustrated in Tables 2.4 and Table 2.5, for the spatial scalability of ratio 1.5 under the intra main configuration, our PCPM method reduces the EL coding execution time by an average of 67.61% compared to unmodified SHVC, at an average cost of 0.29% bitrate increase. PCPM also reduces the average execution time for BL+EL1+EL2 by 48.79%. For the spatial scalability of ratio 2 under the intra main configuration, the PCPM method achieves EL time reduction of 63.95% at a cost of 0.79% increase in BD-BR, on average. Our PCPM reduces the BL+EL1+EL2 total execution time of 51.57%. The CU size prediction accuracy of the PCPM method is more than 90% on average for the intra-main configuration, exactly the same as that of the random access main configuration.

In summary, our proposed CTU prediction scheme significantly reduces the execution time of the quality and spatial scalability with minimal effect on the bit rate/quality of the coded video.

Table 2.5 The impact of the PCPM method on bitrate and PSNR for the spatial scalability under intra main configuration.

Video Sequence	Scalable Ratio	PCPM	
		BD-PSNR (dB)	BD-BR
Traffic	2	-0.010	0.20%
PeopleOnStreet	2	-0.013	0.23%
Kimono	1.5	-0.002	0.05%
	2	-0.007	0.19%
ParkScene	1.5	-0.020	0.47%
	2	-0.017	0.41%
Cactus	1.5	-0.020	0.10%
	2	-0.019	0.57%
BasketballDrive	1.5	0.016	0.47%
	2	0.055	1.87%
BQTerrace	1.5	-0.019	0.38%
	2	-0.034	0.84%
BasketballDrill	2	-0.076	1.52%
BlowingBubbles	2	-0.053	1.13%
BasketballPass	2	-0.088	1.55%
BQSquare	2	-0.013	0.17%
Average	1.5	-0.009	0.29%
	2	-0.025	0.79%

2.4 Conclusions

The focus of this Chapter is on developing complexity reduction schemes for spatial SHVC encoder. First we propose an adaptive search range adjustment for the inter prediction process in spatial SHVC. Then, a quad-tree prediction method at CU level for the spatial SHVC is proposed. The proposed method uses the coding tree unit (CTU) partitioning structures (at the coding unit level) of already encoded CTUs in the base layer, current enhancement layer (EL), and the previous EL (if available) as the predictors for the CTU structures of to-be-encoded CTUs in the current EL. Our scheme utilizes the Bayesian approach to predict the coding unit sizes of the CTUs of the enhancement layers. In order to translate the CTU structures of the predictors into numbers (labels), we introduce a new labeling system. This labeling system

allows us to use a small training dataset, making modeling possible. The resulting model can be fine-tuned during its implementation using a few frames of the incoming video. In this case, the number of learning parameters is reduced significantly compared to the case that the actual CTU based labeling system is used. Our labeling system also makes the training and testing processes faster than those of the actual CTU based labeling system. The performance of the proposed methods was tested over a representative set of video sequences and was compared against the unmodified SHVC encoder as well as the state-of-the-art complexity reduction scheme and combinations. In summary, our final combined scheme outperforms the state-of-the-art method in term of total complexity reduction by 13.20%, on average. It also, reduces the total encoding execution time compared to the unmodified SHVC by more than 39.26% on average, while barely hampering the overall bitrate.

3. Complexity Reduction Methods for Quality Extension of SHVC

For the quality scalability, there are one base layer (BL) and one or more enhancement layer (EL) of different qualities (better than BL). In receiver side the base layer will be decoded and based on the capacity of the channel and the display, zero or one or more number of ELs will be decoded to show 2D contents with better qualities. The ELs and BL contain the same video of different qualities. Therefore, by utilizing the correlation between the BL and ELs we can speed up the process of encoding ELs.

In this Chapter, first we present a content adaptive complexity reduction scheme for quality scalability. This scheme utilizes the correlation between the enhancement layers and the base layer to minimize redundant computations while encoding the enhancement layer. The presented scheme adaptively adjusts the motion search range in the enhancement layer based on the motion vector information of the base layer and implements an adaptive early-termination approach for inter and intra prediction mode search in the enhancement layer.

The complexity of SHVC is mainly because of inter/intra prediction mode search of the coding units. In this regard, we propose three different schemes for mode prediction of quality SHVC. First, we propose a hybrid mode prediction scheme based on statistical studies that are conducted on the training data. The efficiency of this scheme is determined by the correlation between the content and coding configuration of the training and test videos. To reduce this dependency, a content adaptive fast mode assigning method based on the Naive Bayes classifier is proposed which uses the mode information of the corresponding BL CU and the four neighboring blocks in EL for mode prediction. This method (the second mode prediction method) makes a probabilistic method using the training videos. For the test video, the model is fine-tuned using the mode information of the first second of the new scene. In order to improve

the mode prediction accuracy and complexity reduction performance of the second mode prediction method, we propose an online-learning based mode prediction method. The proposed method uses the probabilistic approach and Bayesian classifier to predict the inter/intra modes of CUs in the ELs. Similar to the second mode prediction method, the probabilistic model uses the modes of the corresponding block in the BL and the four neighboring blocks in the EL. However, our online-learning based mode prediction method uses more coding information, i.e. the mode and motion information of the co-located block in the previous frame in the EL, to build the probabilistic model compared to the second mode prediction method. Unlike the Naive based fast mode assigning method, our method uses an RD cost threshold and a probability threshold to improve the prediction accuracy. The big advantage of our proposed scheme is that, unlike the second mode prediction method which does not use online-learning, our scheme uses gradually fine-tuned probabilistic modeling based on content and the quantization parameters. Our proposed scheme finds the mode with the lowest RD-cost for EL CUs by checking a smaller number of modes compared to the second mode prediction method. Unlike Naive based fast mode assigning method which uses unmodified SHVC for the first second of each scene, the proposed scheme is able to start decreasing the complexity much earlier, depending on the video content. Finally, we combine our best mode prediction approach (online-learning based mode prediction) with our content adaptive complexity reduction scheme to achieve higher performance.

The rest of Chapter is organized as follows. In Section 3.1 a content adaptive complexity reduction scheme is presented. In Section 3.2.1, a hybrid complexity reduction scheme is proposed for mode prediction process. Section 3.2.2 presents our Naive based fast mode assigning method proposed for quality SHVC. In Section 3.2.3, online-learning based mode

prediction method is proposed. Experimental results are presented in Section 3.3. Finally, conclusions are given in Section 3.4.

3.1 Content Adaptive Complexity Reduction Scheme for Quality/ Fidelity Scalable HEVC Based on Rate Distortion Prediction

Considering that HEVC is already highly complex, it is inevitable that the scalable extension also to inherit its complexity. Thus, one of the important factors that will lead to a widespread adoption of this emerging standard is reduction of its complexity by addressing cost efficiency and power requirements, which will allow it to be used for a variety of real-time applications.

The focus of this Subsection is to reduce the complexity of SNR/Quality scalable HEVC by minimizing the redundant computations involved in intra and inter prediction process while encoding the enhancement layer. The following Subsections elaborate on the proposed scheme.

3.1.1 Adaptive Search Range Adjustment

In the inter prediction process, selecting a large search range leads to high computational costs, while selecting a small search range produces poor matching results. The optimal motion search range would result in reduced complexity without hampering the compression performance.

In the case of scalable video coding, this may be possible since there is a correlation between the base layer and the enhancement layer, the MVs of the base layer and those of the enhancement layer are also correlated. In our study we utilize this correlation to select the proper motion search range for the enhancement layer based on the motion information of the base layer. Our approach is inspired by the scheme proposed for the existing H.264/SVC standard in

[46]. We classify the CTUs within each frame in the base layer to three different groups: 1) with homogeneous motion, 2) with moderate motion, and 3) with complex motion. This is achieved by defining the motion vector deviation (so called MV homogeneity) of each CTU as follows [46]:

$$MVH_{m,n} = MVHx_{m,n} + MVHy_{m,n}$$

$$MVHx_{m,n} = \frac{1}{T} \sum_{(i,j)\in CTU_{m,n}} \left| mvx_{i,j} - 1/T \sum_{(i,j)\in CU_{m,n}} mvx_{i,j} \right|$$

$$MVHy_{m,n} = \frac{1}{T} \sum_{(i,j)\in CTU_{m,n}} \left| mvy_{i,j} - 1/T \sum_{(i,j)\in CU_{m,n}} mvy_{i,j} \right| \quad (3.1)$$

where T is the total number of MVs assigned to all CUs within the CTU. $MVH_{m,n}$ indicates the motion homogeneity of the CTU. $MVHx_{m,n}$ and $MVHy_{m,n}$ are the horizontal and vertical components of $MVH_{m,n}$. m and n are the coordinates of the CTU, $mvx_{i,j}$ and $mvy_{i,j}$ are the horizontal and vertical components of motion vector of the CU with the coordinates of (i, j), respectively. Once the motion vector deviation of each CTU is available, we can classify the CTUs as follows:

$$MVHC_{CTU} = \begin{cases} MVH_{m,n} < T_1 : \text{CTU} \in \text{region with homogeneous motion} \\ T_1 \le MVH_{m,n} < T_2 : \text{CTU} \in \text{region with moderate motion} \\ MVH_{m,n} \ge T_2 \quad : \text{CTU} \in \text{region with complex motion} \end{cases} \quad (3.2)$$

where $MVHC_{CTU}$ indicates the motion vector homogeneity class of the CTU. T_1 and T_2 are the threshold values defined based on the average motion vector homogeneity (MVH_{ave}) of the whole frame as follows [46]:

$$T_1 = MVH_{ave}\ ,\ T_2 = 0.5 \times MVH_{ave}^2$$

$$MVH_{ave} = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (MVHx_{m,n} + MVHy_{m,n}) \quad (3.3)$$

where *M* and *N* are total number of CTU rows and columns respectively in each frame. In order to design a search range adjustment method, we conduct statistical studies on coding information of our training data. The search range intervals are chosen similar to Section 2.1 in Chapter 2. In this regard, five training video sequences (PartyScene (832×480, 50fps), BQMall (832×480, 60fps), BQSquare (416x240, 60fps), BlowingBubbles (416x240, 50fps), and Vidyo3 (1280×720, 60 fps)) are encoded using SHM 3.0. The quantization parameters (QPs) used for the base layer and enhancement layer (QPB, QPEL) are as follows: (26, 22), (30, 26), (36, 32) and (40, 36). The relationship between the largest MVs of the CTUs in the EL and the motion homogeneity class of the co-located CTUs in the BL is reported in Table 3.1. Note that the statistical information reported in this table is calculated by averaging the results obtained from different QP settings for each video sequence. As can be seen, for the case that the BL CTU belongs to the region with homogeneous motion, the chance of having largest MV that is less than or equal to SR/16 is more than 90%. Similarly, for the case that the BL CTU belongs to the region with moderate motion, the chance of having largest MV that is less than or equal to SR/4 is more than 90%. Based on the results presented in Table 3.1, the motion search range for the co-located CTU in the enhancement layer is adaptively adjusted as follows [100]:

Table 3.1 Motion vector distribution of the EL, given the motion vector homogeneity of the BL.

BL's MVHC	Appropriate SR for EL					
	|MVs|≤SR/32	|MVs|≤SR/16	|MVs|≤SR/8	|MVs|≤SR/4	|MVs|≤SR/2	|MVs|>SR/2
Region with homogeneous motion	75.10%	92.90%	95.00%	95.40%	99.20%	0.8%
Region with moderate motion	33.30%	71.40%	81.90%	90.90%	92.90%	7.1%
Region with complex motion	32.20%	35.70%	39.20%	41.20%	43.30%	56.7%

$$SR' = \begin{cases} round\left(\frac{SR}{16}\right), & CTU \in region\ with\ homogenious\ motion \\ round\left(\frac{SR}{4}\right), & CTU \in region\ with\ normal\ motion \\ SR, & CTU \in region\ with\ complex\ motion \end{cases} \quad (3.4)$$

where SR is the defined motion search range for the base layer and SR' is the adjusted search range of the CTU in the enhancement layer. Note that depending on the class of the CTU in the base layer, the search range of the co-located CTU in the enhancement layer is adjusted, and all the CUs within that CTU will have the same adjusted motion search range setting. As it can be observed from (3.4), the search range can become quite small, depending on the type of the CTU. Taking into account that there might be several CUs (up to 64) within a CTU, this scheme will significantly reduce the computational cost.

3.1.2 Early Termination Mode Search

Note that during inter prediction in HEVC, the encoder goes through all three inter prediction modes, first checking for the skip and merge modes, which are computationally less expensive compared to the explicit motion vector encoding process. Our objective here is to implement early termination (ET) mode-search, so that the encoder does not need to go through all the modes, thus significantly reducing the computational complexity.

The HEVC encoder, in the inter/intra prediction mode selection process, calculates the RD cost for each mode and the one with minimum RD cost is selected. In mode search, if the RD cost of the current to-be-coded CU in the enhancement layer is predicted from the already coded CUs in the base layer and enhancement layer, once the RD cost of a mode is close or equal to the predicted RD cost, the mode search can be terminated. This will significantly reduce the computational complexity. In order to find a prediction for the RD cost of the current CU in the enhancement layer, the RD cost of the already coded CUs in the enhancement layer and that of

their co-located CUs in the base layer is utilized. Figure 3.1 shows an example of the arrangement of the CUs whose information is utilized to predict the RD cost of the to-be-coded CU in the enhancement layer. The neighboring CUs in the enhancement layer are similar to the candidates that HEVC chooses for the merge mode motion search. Inspired by [46], we assume that there is an additive model between the RD cost of the CUs in the enhancement layer and their co-located CUs in the base layer as follows [100], [101]:

$$RDcostE_{C_{predict}} = \left(\alpha_0 \frac{RDcostE_T}{RDcostB_T} + \alpha_1 \frac{RDcostE_L}{RDcostB_L} + \alpha_2 \frac{RDcostE_{TL}}{RDcostB_{TL}} + \alpha_3 \frac{RDcostE_{TR}}{RDcostB_{TR}}\right) * RDcost\ B_c \quad (3.5)$$

where $RDcostE_{Cpredict}$ is the predicted RD cost of current CU in the enhancement layer, $RDcostB_c$ is RD cost of the co-located CU in the base layer, $RDcostE_T$, $RDcostE_L$, $RDcostE_{TL}$ and $RDcostE_{TR}$ denote the RD cost of the four spatial neighbors of the current CU (see Figure 3.1 for the arrangement of CUs), $RDcostB_T$, $RDcostB_L$, $RDcostB_{TL}$ *and* $RDcostB_{TR}$ are the RD cost values of the corresponding CUs in base layer, and α_1, α_2 and α_3 are weighting constants. We compute these weighting constants in the following Subsection.

Once the predicted RD cost for the current CU is available, we define a threshold for early termination of mode search in the enhancement layer as follows:

$$Thr = min(RDcostE_T, RDcostE_L, RDcostE_{TL}, RDcostE_{TR}, RDcostE_{Cpredict}) \quad (3.6)$$

Basically, using this threshold the encoder instead of testing all the modes, it terminates the mode search if the RD cost of a mode is less than the threshold, and selects that mode as the best

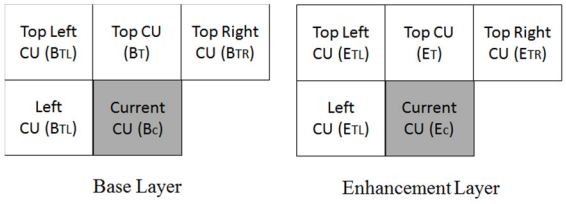

Figure 3.1 Current CU and its four spatial neighbors of base layer and Enhancement layer.

one. Otherwise, it continues testing other modes till this criterion is met. Note that this scheme is applied to the CUs with at least two already-coded neighboring CUs.

In the case where the size of co-located CUs in the base layer is not similar to the one in the enhancement layer, the RD cost of the co-located CU is normalized to its size and the RD cost used in our calculation is updated as follows:

$$RDcost_n = D\frac{w_E \times h_E}{W_B \times H_B} + \lambda_B * B \quad (3.7)$$

where W_B and H_B are respectively the width and height of the co-located CU in the base layer, w_E and h_E is the width and height of the current CU in the enhancement layer, λ_B and B are the Lagrangian constant value and the bit-cost of the co-located CU in the base layer respectively, and $RDcost_n$ is the RD cost value to be used in finding the predicted RD cost and the threshold.

HEVC intra prediction coding tool provides up to 35 directional prediction modes including DC and Planar modes for luma component of each PU. Number of modes which HEVC checks to find the best RD cost depends on the size of the PUs [4]. The threshold defined in (3.6) is also used in intra prediction of the enhancement layer to further reduce the complexity of encoder. Fig. 3.2 provides a block diagram of our proposed complexity reduction scheme.

3.1.2.1 Determining Weighting Constants

In order to find the proper weighting constants in equation (3.5), the Linear Least Square method is used. Our objective is to minimize the difference between the predicted RD cost and the real RD cost of the best mode (without using ET) for the current to-be-coded CU in the enhancement layer. Our objective is formulated as follows:

$$\hat{\alpha} = argmin_{\alpha_i}|(S - S')^2|, \text{ i=0,1,2,3} \quad (3.8)$$

where S is a matrix that contains the real RD cost values of the best modes selected by HEVC for the current CU in the enhancement layer ($RDcostE_C$) divided by $RDcostB_C$, S' denotes a matrix which contains the predicted RD cost of the current CU ($RDcostE_{Cpredict}$)divided by $RDcostB_C$. We can re-write S' as follows:

$$S' = QA = \begin{bmatrix} q_{11}, q_{12}, q_{13}, q_{14} \\ q_{21}, q_{22}, q_{23}, q_{24} \\ . \quad . \quad . \quad . \\ . \quad . \quad . \quad . \\ q_{n1}, q_{n2}, q_{n3}, q_{n4} \end{bmatrix} [\alpha_0, \alpha_1, \alpha_2, \alpha_3]^T \tag{3.9}$$

where

$$q_{i1} = \frac{\mathrm{RDcosE_{Ti}}}{\mathrm{RDcostB_{Ti}}}, \; q_{i2} = \frac{\mathrm{RDcosE_{Li}}}{\mathrm{RDcostB_{Li}}}, \quad q_{i3} = \frac{\mathrm{RDcosE_{TLi}}}{\mathrm{RDcostB_{TLi}}}, \; \mathrm{q_{i4}} = \frac{\mathrm{RDcosE_{TRi}}}{\mathrm{RDcostB_{TRi}}} \qquad \mathrm{i=1,2,3,\ldots,n}$$

Thus, the weighting constants are calculated as follows:

$$A = (Q^T Q)^{-1} Q^T S \tag{3.10}$$

We use a train dataset (five representative video sequences) to calculate the weighting constants. These sequences include: PartyScene (832×480, 50fps), BQMall (832×480, 60fps), BQSquare (416x240, 60fps), BlowingBubbles (416x240, 50fps), and Vidyo3 (1280×720, 60 fps) [92]. We code the video streams, record the real RD cost values, calculate the predicted RD cost based on equation (3.5), and find the weighting constants based on equation (3.10). In the case

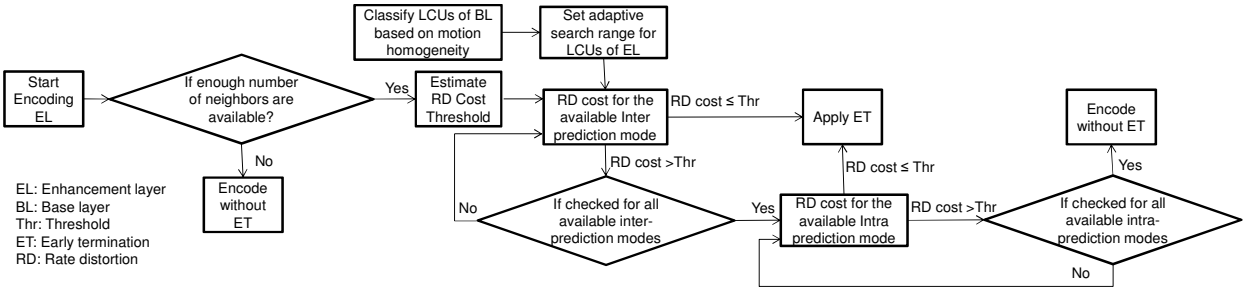

Figure 3.2 Block diagram of our content adaptive complexity reduction scheme.

that all four spatial neighbors (T, L, TL and TR) are available (see Fig. 3.1), the estimated weighting constants are as follows: $[\alpha_0, \alpha_1, \alpha_2, \alpha_3]$ = [0.35, 0.32, 0.16, 0.17]. When $RDcostE_{TR}$ is not available, $[\alpha_0, \alpha_1, \alpha_2, \alpha_3]$ = [0.4505,0.4055,0.1404, 0]. If $RDcostE_{TL}$ is not available – which means that the $RDcostE_L$ is not available either -we use two upper neighbors to predict the RD cost, and the weighting constants are $[\alpha_0, \alpha_1, \alpha_2, \alpha_3]$ = [0.5194, 0, 0, 0.4806]. The weighting constants of the top and left neighboring CUs when available are larger than the others, denoting that they are more correlated with the current CU.

3.2 Mode Search Complexity Reduction Schemes

In Section 3.1 a content adaptive complexity reduction scheme is proposed. In this Section three mode prediction schemes are proposed. In 3.2.1, the first scheme, hybrid complexity reduction scheme, is proposed based on statistical studies which are conducted on the training data. The efficiency of this scheme is determined by the correlation of the content and the coding configuration of the training and the test videos. In order to reduce this dependency, we propose the second scheme based on Bayesian approach. The second scheme, which is presented in Section 3.2.2, creates the initial probabilistic model using the training data. Then, for the test video the first second of the scene is utilized to fine-tune the initial model which reduces the dependency of the probabilistic model on the training data. For the rest of the frames the second scheme uses Naive based classifier. In order to improve the efficiency of the second mode prediction scheme, we propose the third mode prediction scheme. Similar to the second scheme, the third scheme creates initial probabilistic model using the training videos. The third scheme uses online-learning approach which updates the initial model during the course of encoding. Note that the fine-tuning process in the second scheme and online-learning approach updates the

model for the new scene and the encoding configuration. The online-learning mode prediction approach is presented in detail in Section 3.2.3.

3.2.1 Mode Search Complexity Reduction Scheme Based on Statistical Studies

The focus of our study is to reduce the complexity of the SNR/Quality scalable HEVC by minimizing the redundant computations involved in the intra and inter prediction process. This scheme aims at achieving real-time coding and transmission of several quality versions of the same video content, allowing for cost effective universal access of digital media. In summary, in this work we first introduce two different and independent complexity reduction methods, followed by a hybrid method that consists of a suitable combination of those two methods that results in the best possible performance. First, two EL early mode prediction methods are proposed; one of them utilizes correlations between different CUs belonging to the same quad-tree structure in each EL CTU and the other one uses the correlation between the mode of EL CUs and the mode information of the corresponding CUs in the reference layer to predict the mode of the to-be-coded CU in the EL. Note that EL represents two enhancement layers, EL1 and EL2. Finally, a more advanced complexity reduction scheme is proposed that appropriately combines the two early mode prediction methods into a hybrid method, leading to the best performance overall. The following Subsections elaborate on our proposed schemes.

3.2.1.1 EL Early Mode Prediction Methods

One way of reducing the complexity of SHVC is to predict which modes are more likely to be the best modes for the CUs in EL and only check those modes [102]. To this end, we introduce two mode prediction (MP) methods described in the following Subsections.

3.2.1.1.1 EL Early Mode Prediction (MP) Method Based on Quad-tree Mode Information

SHVC, inheriting the quad-tree structure of HEVC, first divides each frame into several CTUs of the same size. In this study, the random-access main configuration of SHVC is utilized and the CTU size is set to 64×64. Each 64×64 CTU is subdivided into four 32x32 CUs in the first depth layer of the quad-tree structure [4], [5]. Here we consider the CTU to be the parent of all the CUs in the first depth layer. In the second depth layer of the quad-tree, each CU of the first depth layer (called parent) can be split into four CUs (called children). Accordingly, a CU in the second depth layer of the quad-tree can be split into four CUs in the third depth layer. Fig. 3.3 illustrates a CTU and its four-layer quad-tree structure. As it is observed in Fig. 3.3, the CUs at each depth level are part of the parent CU in the lower depth level. This relationship leads to high correlation between the information of the children CUs and the parent CUs (the blue-color CU in Fig. 3.3 is correlated with tan-color CUs). Note that we call all the corresponding CUs in the previous depth layers of the same CTU the *parent CUs for each to-be-coded CU*. In this approach we focus on decreasing the computational complexity of encoding CUs in the EL based on the quad-tree mode information. A representative set of video sequences including the five different videos (our training sequences) are encoded using the SHVC reference software (SHM. 6.1) with random access main configuration (Hierarchical B pictures, GOP size of 8, SAO, and RDOQ are enabled). In our study, we have two ELs (enhancement layer 1 (EL1) and enhancement layer 2 (EL2)) in addition to the BL. The quantization parameters (QPs) used for the base layer and enhancement layers (QPB, QPEL1, QPEL2) are as follows: (26, 22, 18), (30,

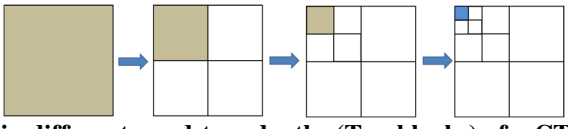

Figure 3.3 The parent CUs in different quad-tree depths (Tan blocks) of a CTU, which can be used for predicting the mode of the first CU at depth 3 (Blue block).

26, 22), (34, 30, 26) and (38, 34, 30). These training video sequences include: PartyScene (832x480, 50fps), BQMall (832x480, 60fps), BlowingBubbles (416x240, 50fps), and Vidyo3 (1280x720, 60 fps). During encoding the training video sequences, the mode information (with the lowest RD cost) of all the CUs belonging to the same quad-tree in the EL is recorded. This information is available as the SHVC encoder goes through all possible modes for all possible quad-tree structures in an EL CTU. In order to decrease the complexity of encoding ELs, we use our statistical results to suggest most probable mode candidates for EL CUs. Our studies show that in case that the merge [5] mode is the mode with the lowest RD cost for a parent CU, the four most probable modes are the merge, Inter NxN, Inter NxN/2, and Inter N/2xN modes [5] for the child CUs. Note that Inter NxN/2 and Inter N/2xN modes can use the inter layer reference picture (ILRP) in addition to temporal reference picture [60]. Here, NxN indicates the size of current CU. Table 3.2 shows the average probability of selecting the four most probable modes (merge mode, Inter NxN, Inter NxN/2, and Inter N/2xN) for the child CUs of the EL (EL1 or EL2) at different depth layers of quad-tree structure if the mode with the lowest RD cost for at least one of the parent CUs in previous depth layers is merge mode. Note that the average statistical data over different QP settings and training video sequences are reported. For the EL (EL1 and EL2) child CUs at depth 1, there is more than 91% chance that the mode with the lowest RD cost is merge mode or Inter NxN or Inter NxN/2 or Inter N/2xN, if the merge mode is the mode with lowest RD cost for the parent CU. In addition, Table 3.2 shows that for the EL1 children CUs at depth larger than 1 and for the EL2 children CUs at depth 3, the probability of having merge mode as the mode with the lowest RD cost is more than 95%, if the mode of the parent CU is merge mode. Moreover, Table 3.2 shows that the cumulative probability of having

Table 3.2 Average probability of using merge or Inter NxN or Inter N/2xN or Inter NxN/2 mode for coding child CUs when merge mode has the lowest RD cost for at least one of the parent CUs.

Layer	Mode #1	Average Probability of observing the mode #1 for the CU in the depth layer d when Merge mode has the lowest RD cost for the corresponding CU in the depth layer k					
		(k,d)=(0,1)	(k,d)=(0,2)	(k,d)=(0,3)	(k,d)=(1,2)	(k,d)=(1,3)	(k,d)=(2,3)
EL_1	Merge	0.846	0.950	0.976	0.973	0.983	0.987
	Inter NxN	0.022	0.009	0.009	0.005	0.005	0.004
	Inter N/2xN with ILRP	0.020	0.008	0.004	0.004	0.003	0.002
	Inter NxN/2 with ILRP	0.023	0.011	0.006	0.006	0.005	0.004
	Sum	0.912	0.979	0.996	0.989	0.996	0.998
EL_2	Merge	0.810	0.891	0.962	0.891	0.971	0.966
	Inter NxN	0.028	0.024	0.016	0.026	0.013	0.013
	Inter N/2xN with ILRP	0.037	0.021	0.007	0.019	0.005	0.005
	Inter NxN/2 with ILRP	0.046	0.019	0.008	0.023	0.007	0.010
	Sum	0.921	0.955	0.992	0.959	0.996	0.994

merge mode or Inter NxN mode is more than 91% for the EL2 child CU at depth 2, if the merge mode is the mode with the lowest RD cost for the parent CU.

Our studies show that in the case that the mode with the lowest RD cost for the parent CU is either the NxN inter layer reference (ILR) mode (with uni-prediction) [60] or Intra NxN [5], then the most probable modes are ILR NxN, Merge and Intra NxN modes for EL CU. Note that in ILR NxN mode (with uni-prediction), only the inter layer reference picture is utilized to predict the current EL [60], [103]. Table 3.3 shows the cumulative probability of the ILR NxN or Intra NxN or Merge mode having the lowest RD cost for an EL (EL1 or EL2) CU when the ILR NxN mode or Intra NxN mode has the lowest RD cost for at least one of its parent CUs in previous depth layers is more than 90%. Note that Table 3.3 reports the average statistical data over different QP settings.

Based on the statistical results obtained from our experiments, we propose a mode prediction method for CUs in the EL based on the quad-tree mode information. Fig. 3.4 illustrates the block diagram of our proposed quad-tree based early termination (MP) method. In this approach, based on the depth and scalable layer few modes are suggested for the current to-be-coded CU in the EL, if the lowest RD cost corresponds to the merge mode for at least one of the parent CUs in the

previous depth layers of the same quad-tree structure. In this case, the rest of modes won't be tested for the current CU. If the quad-tree depth of the to-be-coded EL (EL1 and EL2) CU is one and the merge mode has the lowest RD cost for at least one of the parent CUs in the same quad-tree, our method checks only the merge mode, Inter NxN, Inter N/2xN, and Inter NxN/2 for the current to-be-code CU. In the case that the quad-tree depth of the to-be-coded EL1 CU is larger than one and the merge mode is chosen for at least one of its parent CUs as the mode with the lowest RD cost, our method only checks the merge mode for the to-be-coded EL1 CU. In the case that the quad-tree depth of the to-be-coded EL2 CU is two and merge mode has the lowest RD cost for at least one of the parent CUs in the same quad-tree, our method checks only the merge mode, and Inter NxN for the current to-be-code CU. If the quad-tree depth of the to-be-coded EL2 CU is three and merge mode has the lowest RD cost for at least one of the parent CUs in the same quad-tree, our method checks only merge mode. Otherwise, if Intra NxN or ILR

Table 3.3 Probability of using ILR NxN or Merge or Intra NxN mode for coding child CUs when ILR NxN or Intra NxN mode has the lowest RD cost for at least one of the parent CUs.

Layer	Mode #1	Mode #2	Average Probability of observing the mode #2 for the CU in the depth layer d when mode #1 has the lowest RD cost for the corresponding CU in the depth layer k					
			(k,d)=(0,1)	(k,d)=(0,2)	(k,d)=(0,3)	(k,d)=(1,2)	(k,d)=(1,3)	(k,d)=(2,3)
EL_1	Intra NxN	ILR NxN	0.505	0.470	0.550	0.385	0.528	0.456
		Merge	0.358	0.370	0.345	0.342	0.361	0.431
		Intra_NxN	0.094	0.140	0.089	0.218	0.091	0.070
		Sum	0.957	0.980	0.984	0.945	0.981	0.957
EL_1	ILR NxN	ILR NxN	0.615	0.651	0.545	0.802	0.779	0.831
		Merge	0.271	0.232	0.343	0.107	0.155	0.110
		Intra_NxN	0.037	0.092	0.081	0.037	0.038	0.034
		Sum	0.923	0.975	0.970	0.946	0.972	0.976
EL_2	Intra NxN	ILR NxN	0.512	0.482	0.497	0.402	0.496	0.386
		Merge	0.225	0.293	0.386	0.355	0.377	0.407
		Intra_NxN	0.214	0.168	0.094	0.202	0.107	0.166
		Sum	0.951	0.943	0.976	0.959	0.981	0.959
EL_2	ILR NxN	ILR NxN	0.610	0.560	0.509	0.769	0.730	0.782
		Merge	0.268	0.293	0.376	0.122	0.158	0.133
		Intra_NxN	0.043	0.066	0.076	0.080	0.082	0.067
		Sum	0.921	0.919	0.961	0.971	0.970	0.982

NxN has the lowest RD cost for at least one of the parent CUs in the same quad-tree, our method checks only ILR NxN mode, Intra NxN, and the merge mode for the current to-be-code EL CU. Note that the encoder starts from the largest size CU (quad-tree depth equal to zero) and checks all the available modes to find the mode with the lowest RD cost. Then, for the next quad-tree depth layer our quad-tree EL early mode prediction method is utilized to predict the mode. In case that none of the parent CUs have a merge or ILR NxN or intra NxN as the mode with the lowest RD cost, all the available modes will be checked. This process continues until the quad-tree partitioning reaches the smallest size CU.

Note that if the depth layer is equal to zero, our method checks all the available prediction modes to find the best mode. For each depth layer, other than the first, our method uses the mode information of the parent CUs to predict the mode for the children CUs. Note that our method is not able to predict the mode if the parent CUs have a mode other than the merge mode, Intra NxN mode, or ILR NxN mode. In the case that our MP method is not able to predict the mode with the lowest RD cost for the to-be-coded EL CU, it checks all the avialable modes for that CU.

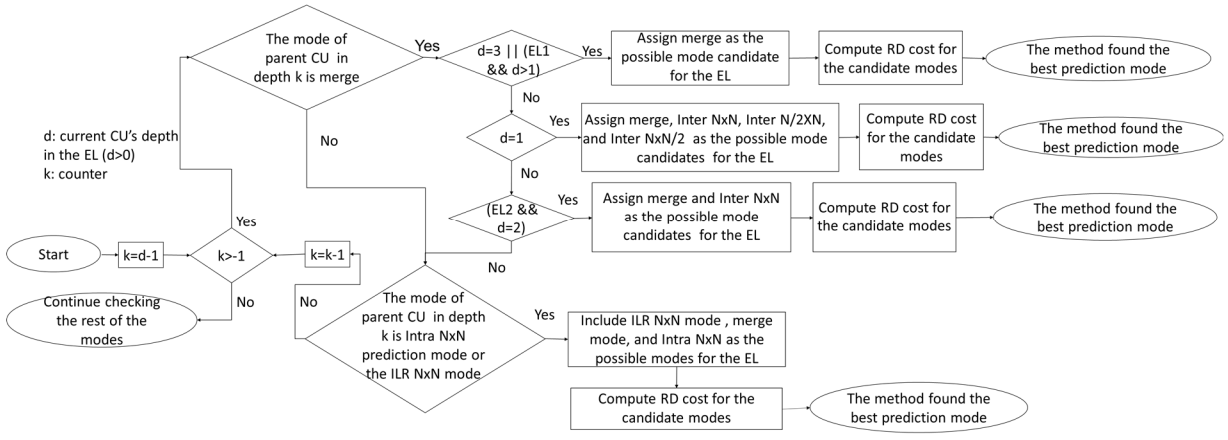

Figure 3.4 Flowchart of the proposed quad-tree based MP method.

3.2.1.1.2 EL Early Mode Prediction Based on Reference Layer's mode

In addition to using the quad-tree structure for mode prediction, the correlation between the mode information of the CUs in the EL and the ones in the reference layer may also be used. In this approach, we first verify the existence of such correlation by analyzing the results obtained by encoding our training video set. These videos were encoded using SHVC reference software (SHM 6.1) with random access main configuration (Hierarchical B pictures, GOP size of 8, SAO, and RDOQ are enabled). The QP used for the base layer, EL1, and EL2 (QPBL, QPEL1, QPEL2) are as follows: (26, 22, 18), (30, 26,22), (34, 30,26) and (38, 34, 30).

To verify the correlations between the mode of CUs in the EL and the mode of the corresponding (co-located) CUs in the reference layer, a set of representative video sequences (our training video dataset) are encoded using SHVC and the mode information (with the lowest RD cost) of all the CUs in BL, EL1, and EL2 is recorded. Note that the BL is the reference layer for the EL1. On the other hand, the BL and the EL1 are the reference layers for the EL2. In our method, all the information of mode selection in BL is saved in order to be used for mode

Table 3.4 Average probability of observing the Merge mode or the Inter NxN for EL CU when the Merge mode has the lowest RD cost for the reference layer CU at the same depth layer (with the same size) as the EL CU.

Current layer - Reference layer	mode #1	Average Probability of observing mode #1 for a CU in the current scalable layer when the merge mode has the lowest RD cost for the corresponding CU in the reference layer (in depth layer *d)*			
		d=0	d=1	d=2	d=3
EL_1-BL	Merge	0.85	0.88	0.93	0.97
	Inter NxN	0.04	0.03	0.01	0.01
	Sum	0.90	0.92	0.95	0.98
EL_2-BL	Merge	0.74	0.85	0.92	0.96
	Inter NxN	0.06	0.03	0.02	0.02
	Sum	0.80	0.89	0.94	0.97
EL_2-EL1	Merge	0.83	0.88	0.93	0.97
	Inter NxN	0.07	0.03	0.02	0.01
	Sum	0.90	0.91	0.95	0.99

prediction when the size of the co-located CU in the BL and the to-be coded CU in the EL (EL1 or EL2) are not similar. In our study, we utilize the co-located CU in the reference layer at the same CTU depth layer as the to-be-code EL CU to find the correlations between the BL and the ELs.

Table 3.4 shows the average probability of selecting the merge mode and Inter NxN for the CUs of the EL (EL1 and EL2) at different depth layers of quad-tree structure if the mode with the lowest RD cost for the corresponding CU in the reference layer is the merge mode. Note that the average statistical data over different QP settings and training video sequences are reported. For each CU in the EL, the mode information of the corresponding CU (with the same size) in the reference layer is utilized to find the probabilities. Note that in the conditional probabilities reported in this study '|' means given. The statistical results show that (see Table 3.4):

1- Average probability P(coding EL1 CU at depth smaller than two, using Merge mode or Inter NxN mode | merge mode has the lowest RD cost for the corresponding CU in the BL at same CTU depth layer as the EL1 CU)≥0.9

2- P(coding EL (EL1 or EL2) CU at depth larger than one, using Merge mode | merge mode has the lowest RD cost for the corresponding CU in the reference layer at same CTU depth layer as the EL CU)≥0.9

3- P(coding EL2 CU at depth smaller than two, using Merge mode or Inter NxN mode | merge mode has the lowest RD cost for the corresponding CU in the EL1 at same CTU depth layer as the EL2 CU)≥0.9

Table 3.5 Average probability of observing different modes in EL given the mode of the reference layer CU at the same depth layer (with the same size) as the EL CU.

Mode2	Probability of observing the mode #2 for the CU in the current scalable layer when mode #1 has the lowest RD cost for the corresponding CU in the reference layer						
	ref layer =BL, current layer =EL_1		ref layer =BL, current layer =EL_2		ref layer =EL_1, current layer =EL_2		
	Mode #1=Intra NxN	Mode #1=Intra N/2xN/2	Mode #1=Intra NxN	Mode #1=Intra N/2xN/2	Mode #1=Intra NxN	Mode #1=Intra N/2xN/2	Mode #1=ILR NxN
ILR NxN	0.642	0.671	0.617	0.664	0.411	0.292	0.829
Merge	0.295	0.302	0.294	0.297	0.352	0.529	0.070
Intra NxN	0.042	0.006	0.067	0.021	0.160	0.032	0.058
Intra N/2xN/2	0.002	0.004	0.003	0.006	0.007	0.049	0.003
Sum	0.981	0.983	0.981	0.987	0.930	0.902	0.959

However, for depth 0 and depth1, P(coding EL2 CU using Merge mode or Inter NxN mode| Merge mode has the lowest RD cost for the corresponding CU in the BL at same CTU depth layer as the EL2 CU)<0.9. Therefore, in these cases if the merge mode is the mode with the lowest RD cost for the BL CU, based on our statistical studies we cannot suggest only Merge mode and Inter NxN mode as the mode candidates for the EL2 CU.

In the next step, we investigate other modes. Table 3.5 shows the relationship between the most probable modes (Merge, ILR NxN, and Intra modes [5]) of the EL CUs when Intra NxN or Intra N/2xN/2 [5] or the ILR NxN mode is the mode with the lowest RD cost for corresponding CUs in the reference layer. For each CU in the EL, the mode information of the corresponding reference layer CU (with the same size) is utilized to find the probabilities. More specifically, Table 3.5 shows the average probability of selecting the merge mode or ILR NxN mode or each intra prediction modes for coding the CUs in the EL, given the mode (ILR NxN and Intra) of the corresponding (same size) CUs in the reference layer. Similar to Table 3.4, the reported data in Table 3.5, is average statistical results over different QP settings. The statistical results show that in case that the corresponding CU in the BL is encoded using the Intra NxN mode, the probability of using ILR NxN mode or Merge mode for the EL1 CU (or EL2 CU) is more than 90% at the same CTU depth layer as the BL CU. In the case that the Intra N/2xN/2 mode has the

lowest RD cost for the BL CU, then the probability of having the ILR NxN mode or the Merge mode for the EL (EL1 and EL2) CU (at the same CTU depth layer as the BL CU) is more than 96% on average. The statistical results also show that

1- P(coding EL2 CU using ILR NxN mode or Merge mode or Intra NxN mode| ILR NxN mode or Intra NxN mode has the lowest RD cost for the corresponding CU in the EL1)>0.9

2- P(coding EL2 CU using ILR NxN mode or Merge mode or Intra NxN mode or Intra N/2xN/2 mode | Intra N/2xN/2 mode has the lowest RD cost for the corresponding CU in the EL1)>0.9.

Note, in our studies same probability patterns were observed for CUs located at different coding depths. Therefore, in table 3.5 the average probabilities over CUs at different depths are reported.

Based on the above observations, we propose a new mode prediction method for the EL. In our implementation, if the mode of the BL block at depth 0 or depth 1 is the merge mode, only the merge mode and Inter NxN mode will be checked for the corresponding EL1 CU. Our method suggests the merge mode for EL (EL1 and EL2) CU at depth larger than one, if the merge mode is the mode with lowest RD cost for the corresponding BL CU. For the cases in which the corresponding CU in the BL is encoded using one of the intra prediction modes, only the ILR NxN mode and the Merge mode will be checked for the EL CU. For the cases in which the Merge mode is the lowest RD cost mode for the EL1 CU at depth 0 or depth 1, only the merge mode and Inter NxN is checked for the corresponding EL2 CU. In our method, if the merge mode is the lowest RD cost mode for EL1 CU at depth 2 or depth 3, only the merge mode is suggested for the corresponding EL2 CU. Moreover, if the ILR NxN mode or the Intra NxN is the lowest RD cost mode of the corresponding block in the EL1, only the ILR, Merge, and the

Intra NxN modes are checked for the current block in the EL2. For the cases in which the corresponding block in EL1 is encoded using Intra N/2xN/2, only the ILR NxN, merge, and all the intra prediction modes will be checked for the current block in the EL2.

Note that if the corresponding CU in the BL has a mode other than merge and Intra modes, the EL early mode prediction based on the reference layer's mode is not able to suggest the mode for EL1 CU and EL2 CU based on BL's mode information. In addition, if the corresponding CU in the EL1 has a mode other than the ILR NxN mode, the merge mode, and the intra modes, our method cannot suggest a mode for EL2 CU. Therefore, in the cases in which our method is not able to suggest a mode for an EL, all the available modes are checked for that EL. The block diagram of the proposed MP method based on the reference layer's mode information is shown in Fig. 3.5.

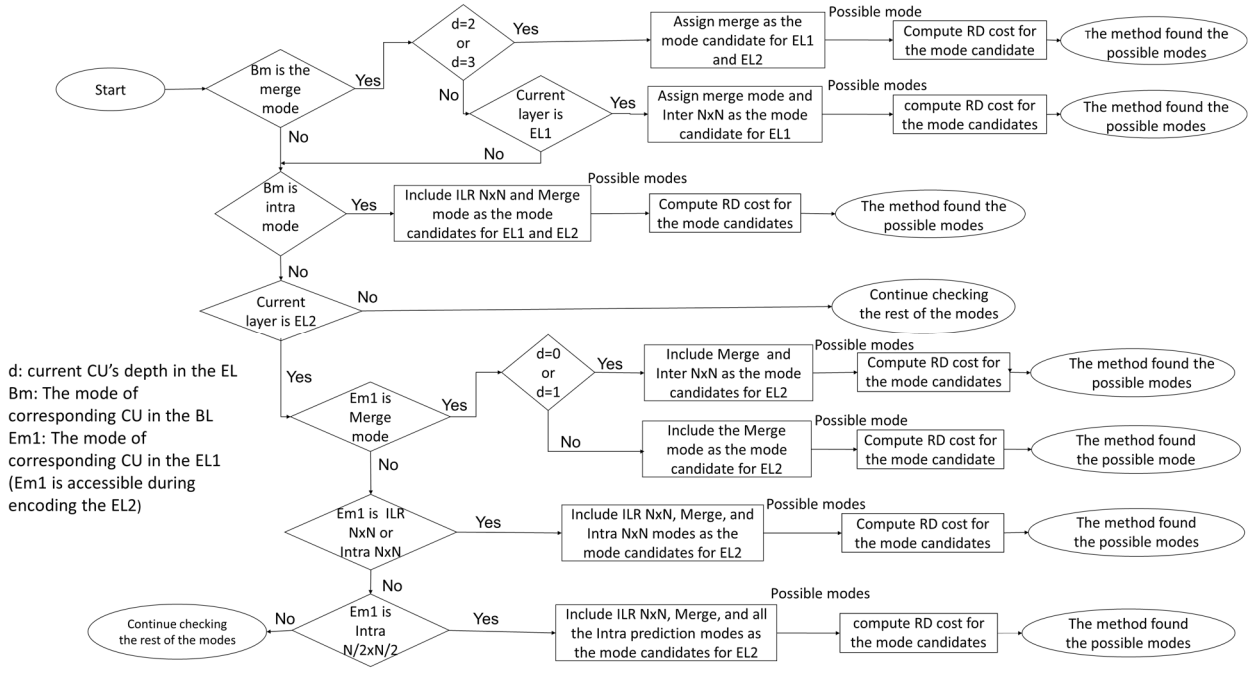

Figure 3.5 Flowchart of the EL Early MP method based on reference layer's mode.

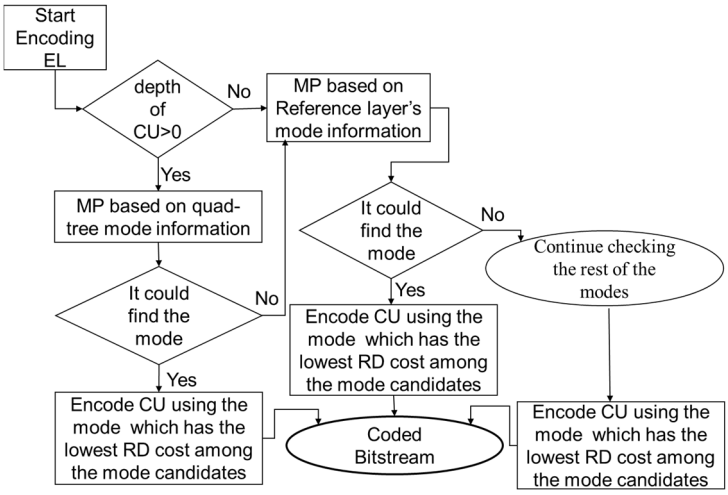

Figure 3.6 Block diagram of our proposed hybrid complexity reduction scheme for SHVC.

3.2.1.2 Hybrid Complexity Reduction Scheme Based on Statistical Studies

In the previous Subsections we present two independent complexity reduction methods. Each of these methods partially utilizes different redundancies/correlation within the CUs in the EL as well as between the EL and BL. In our final step, we design a hybrid complexity reduction scheme, which utilizes a suitable combination of the above methods.

Fig. 3.6 shows the block diagram of our hybrid complexity reduction scheme. As it is observed, first the BL is encoded using the original/unmodified SHVC. Then during the encoding process of EL (EL1 and EL2), if the depth of the CU is equal to zero, the MP method based on the reference layer's mode information is used for mode prediction. Note that the MP based on the reference layer's mode information is not able to suggest the mode for EL (EL1 and EL2) CU based on BL's mode information, if the corresponding CU in the BL has a mode other than merge and Intra modes. In addition, if the corresponding CU in EL1 has a mode other than the ILR NxN mode, the merge mode, and the intra modes, MP based on the reference layer's mode information cannot suggest a mode for EL2 CU. If this method fails to predict the mode,

the unchecked modes will be checked. Afterwards, the mode, which has the lowest RD cost will be selected as the best mode. If the quad-tree depth layer is greater than zero, our hybrid complexity reduction scheme uses the quad-tree-based MP method to predict the mode of current CU in the EL. Note that the quad-tree based MP method is applicable when the CTU depth layer is greater than zero and the merge mode or Intra NxN mode or ILR NxN mode has the lowest RD cost for parent CUs. If the quad-tree-based MP method fails to suggest the best mode, our scheme uses the of the reference layer's mode information. If this method fails to suggest the best mode, the unchecked modes are checked. Finally, the mode, which has the lowest RD cost will be selected as the best mode for the current CU in the EL.

3.2.2 Naive Bayes Fast Mode Assignment

In the previous Section, a hybrid complexity reduction method was proposed based on statistical studies. This method was only able to predict modes for current EL CU in the cases that the parent CUs or the reference layer CUs are encoded using Merge mode or intra modes or ILR mode. Since the hybrid complexity reduction is designed based on the statistical studies, the correlation between the content and coding configuration of the training and the test videos varies depending on the size of the training dataset and thus does not always achieve the desired accuracy. To reduce this dependency, the focus of this Subsection is to design a content adaptive mode prediction method based on machine learning approach by utilizing the correlation between the base layer and enhancement layer. As mentioned earlier, the SHVC standard utilizes several inter and intra prediction modes to achieve high compression performance. Considering in the inter/intra prediction mode-selection process, the encoder is required to calculate the RD cost for each mode to find the best mode with minimum RD cost, mode

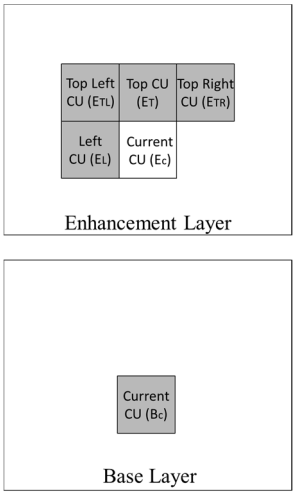

Figure 3.7 Current CU (white block) and its five predictors (Gray blocks).

decision is one of the most computationally involved procedures in a HEVC-based encoder. In the case of SHVC, since there is a high correlation between the base layer and the enhancement layer, the CU modes of the frames in the base layer can help us to speed up the process of selecting modes for the corresponding enhancement frames. Also, the modes of the already encoded neighboring CUs in the enhancement layer are valuable for predicting the mode of the current CU. Therefore the modes of the neighboring CUs in the enhancement layer and the corresponding CU in the base layer are used to predict the mode of the current CU. These five predictors are called predictor CUs hereafter. Fig. 3.7 shows an example of the current CU, its corresponding base layer CU and its four neighbors. E_C indicates the current CU and E_L(left), E_{TL}(top left), E_T(top) and E_{TR}(top right) are its four spatial neighbors whose information is exploited to predict a mode for E_C. The neighboring CUs in the enhancement layer are similar to the candidates that HEVC chooses for the merge mode.

The objective here is to implement a fast mode assigning (FMA) mode-search, so that the encoder does not need to go through all the modes, thus significantly reducing the computational complexity. To this end we approximate a function from the predictor CUs to the current CU or, equivalently, a posterior probability of the mode of current CU given the mode of its predictor

CUs. The estimation of this posterior probability (or equivalently this function approximation problem) can be modeled as a supervised learning problem. This supervised learning problem consists of two stages; the first stage is the training process and the second stage is the test process. During the training process, the encoder encodes the BL using the unmodified SHVC [5]. The SHVC encoder, in the inter/intra prediction mode selection process, calculates the RD cost for each mode and the one with minimum RD cost is selected. Then, the EL is also encoded using the conventional SHVC. In this process all of the available inter-intra modes are checked to find the lowest rate distortion cost. For each CU in the EL, the information about the mode chosen by the encoder is stored. Based on this information, the probability of each mode in the current CU in EL (given the predictors modes) is updated. These conditional probabilities will then be used in the test process. The second stage is the test process. In this stage, first the program encodes the BL using unmodified SHVC, then the encoder starts encoding the EL. The information stored during encoding the BL and the previously encoded CU in the EL is used to estimate a mode for the to-be-encoded CU in the EL, without the encoder being required to check all of the inter-intra prediction modes [104].

As mentioned above, a probability can be assigned to each mode for the current CU given the modes of its predictor CUs. To define this posterior probability, different numbers are assigned to different HEVC modes (inter and intra modes), and each mode is considered as a class. Assume Y is the random variable corresponding to the mode of the current CU, and X is a random vector corresponding to the modes of its predictor CUs. In the case, where there is M different modes in HEVC, the random variable Y can have M different values. Regarding the predictor vector X, if there are L predictor CUs, the length of vector X will be equal to L and each of its components can take M possible discrete values (M possible modes). This results in

M^L-1 different possible values for the random vector *X*. The posterior probability *P(Y|X)*, which in our case indicates the probability of the modes of the predictor CUs given each mode of the current CU in EL, can be calculated using the Bayes rule as follows:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{3.11}$$

where *P(Y)* is the prior probability of the mode of the to-be encoded CU, *P(X|Y)* is the class-conditional density, which defines the distribution of the data that is expected to be seen in each class. The learning algorithm needs to estimate *M-1* different parameters to estimate *P(Y)*, because the probability should sum to one. However, estimating *P(X|Y)* requires learning of an exponential number of parameters, which is an intractable problem [89], [90]. Thereby, the key to use Bayes rule is to specify a suitable model for *P(X|Y)*. In our study to solve the above-mentioned intractability problem, Naive Bayes classifier [90] has been used. The Naive Bayes classifier dramatically reduces the complexity of estimating *P(X|Y)* by making a conditional independence assumption. This learning algorithm assumes that different components of the *X* vector are independent given *Y*. Taking into account the conditional independence assumption we have:

$$P(X|Y) = P(X_1, X_2, \dots, X_L|Y) = \prod_{l=1}^{L} P(X_l|Y) \tag{3.12}$$

Therefore,

$$P(Y|X) = \frac{\prod_{l=1}^{L} P(X_l|Y)\, P(Y)}{P(X)} \tag{3.13}$$

According to the optimal Bayes decision rule [89], [90], the mode of the posterior probability distribution is the predicted mode of the current CU. Therefore, for classifying a new *X*, the following formula can be used:

$$\mathrm{y_m} = \underset{y_m}{\operatorname{argmax}}\, P(Y = y_m) \prod_{l=1}^{L} P(X_l|Y = y_m) \tag{3.14}$$

where y_m is the m^{th} possible value of Y. The normalization part, i.e., $P(X)$, of the posterior distribution has been omitted due to the fact that the denominator does not depend on y_m. The resulting y_m is the predicted mode for the current to be encoded CU.

To find the optimal value of y_m in equation (3.14), we need to have $P(X|Y)$ and $P(Y)$. These probabilities are computed during the training process. A very popular method to estimate these probabilities is the MLE [89]. As explained in Chapter 2, a major drawback of MLE is that when MLE is used for estimating the probabilities, there are some situations in which we have not seen some states (modes) in the training set. Therefore, in this case the classifier overfits and will have problem during the test process [89]. To resolve this problem MAP [91] estimation is employed in this study. MAP estimate resolves the above-mentioned problem by incorporating a prior distribution over the parameter that we want to approximate. In order to facilitate MAP estimation, it is required to assign appropriate prior distribution for the parameters. As a result, the solution to the MAP estimate for $P(Y)$ is:

$$P(Y = y_k) = \frac{N_k + \alpha_k}{Total\ number\ of\ tries + \sum_{k=1}^{M} \alpha_k} \tag{3.15}$$

where α_k determines the strength of the prior assumptions relative to the observed data and M is equal to the number of different values which Y can take. The element N_k is the number of observed instances of class y_k. That is, N_k indicates the number of times the modes of the current CU is equal to y_k. On the other hand, the estimate for $P(X|Y)$ is as follows:

$$P(X_l = x_{lm} | Y = y_k) = \frac{N_{lmk} + \alpha_{lm}}{Total\ number\ of\ tries + \sum_{m=1}^{M} \alpha_{lm}} \tag{3.16}$$

where α_{lm} determines the strength of the prior assumptions relative to the observed data and M is equal to the number of distinct values which X_l can take. The element N_{lmk}denotes the number of times $X_l=x_{lm}$ has been observed in the instances of class y_k [89], [90]. That is, N_{lmk} indicates the number of times the modes of the l^{th} predictor is equal to x_{lm}, while the mode of the current

CU is equal to y_k. To find the hyper parameters α_k and α_{lm}, four representative video sequences are used in our approach. These sequences include: PartyScene (832×480, 50fps), BQMall (832×480, 60fps), Racehorse (832×480, 30fps), and Vidyo3 (1280×720, 60 fps) [92]. These video sequences are different from the video sets used to test our approach.

To implement Naive Bayes FMA (NB-FMA), the first second (exp., 30 frames if the frame rate is 30fps) of the video is coded based on non-modified SHVC [104]. The coding information (modes) of these frames is used for the training process. During the training process the program updates the probabilities. Then, for finding the modes of the rest of the frames the fast mode assigning method is applied. In this stage, first the program encodes the BL. Then, the encoder starts encoding the EL. Unlike the training process, the encoder does not check all of the inter-intra prediction modes. Instead the information of the predictor CUs is used for predicting the mode of the to-be-encoded CU. In this study, the two mode candidates with the highest probability among all modes are chosen, and the encoder calculates the RD cost for these two candidates and chooses the one with the smallest RD cost is selected. The block diagram of the overall method is shown in Figure 3.8.

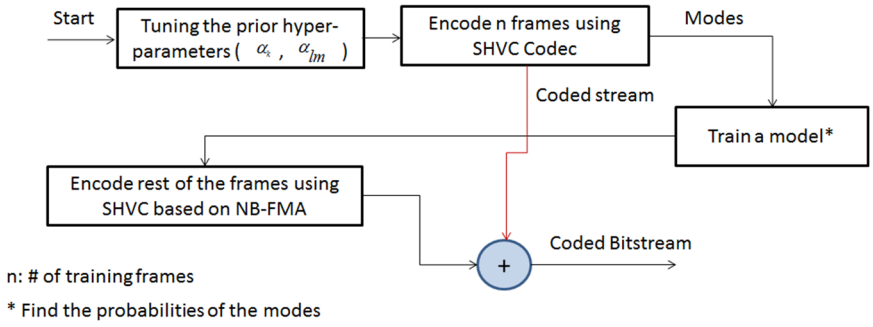

Figure 3.8 Block diagram of the proposed method.

3.2.3 Online-learning Bayesian Based Complexity Reduction Scheme for Quality SHVC

The mode search process of SHVC encoder includes the Skip, Merge mode, several inter and intra prediction modes that are checked for each CU, making it the most time-consuming part of the encoding procedure. In order to decrease the number of mode searching steps utilized by SHVC, it is required to design a scheme that avoids checking every possible option in order to find the inter/intra prediction mode with the lowest RD cost.

The focus of our study is to reduce the complexity of SNR/Quality scalable HEVC (SHVC) by minimizing the redundant computations involved in intra and inter prediction process while encoding the enhancement layers. The objective here is to implement a fast mode assigning (FMA) mode-search, so that the encoder does not need to go through all the modes, thus significantly reducing the computational complexity.

To achieve this goal, proposed method employs a probabilistic classifier to predict the mode of the to-be-encoded CU in EL. The probabilistic classifier uses the quad-tree mode information and the coding information of the already encoded CUs in the BL and the ELs for mode prediction (Section 3.2.3.1).

Then a fast mode assigning method is proposed which utilizes the probabilistic classifier to predict the mode that is more likely to be the inter/intra mode with the lowest RD-cost for the EL CUs (Section 3.2.3.2). Note that we call the mode, which has the lowest RD cost (among all the available modes) as the lowest RD cost (LRC) mode, here after.

3.2.3.1 Probabilistic Classifier

A probabilistic classifier is a classifier that identifies the category (class (Y)) of a new observation vector (X) based on a conditional distribution ($P(Y|X)$). In this Section, Y is the random variable corresponding to the mode of the current CU and X is a random vector corresponding to the predictor features. Note that we call $P(Y|X)$ as the posterior probability. The first step in designing our classifier is to choose appropriate features as the predictors for the posterior probability (probabilistic model). There is a high correlation between two consecutive frames, especially if there is no or limited motion in the scene. In our study, we investigated the correlation between the coding information of the collocated CU in the previous frame and the LRC mode of to-be-coded EL CU as motion of the scene changes. It is observed that in the case the collocated CU in the previous frame is encoded using the merge/skip mode and it belongs to the region with homogeneous motion, the chance of selecting the merge/skip mode for the to-be-coded EL CU is more than 80% (for our training data). Therefore, in order to use this correlation and even find more complex correlations between the LRC mode of to-be-coded EL CU and its collocated CU in the previous frame, we use the motion information and LRC mode of the collocated block in the previous frame of the same EL as the first and the second predictor features (for the LRC mode of to-be-coded EL CU). Note that we call the co-located CU in the previous frame as the temporal predictor CU. In order to use the motion information of temporal predictor CU as the feature, it is required to quantify the motion information of that predictor CU. In this regard, we use the approach suggested in Section 3.1.1 [46], [100] that classifies each CTU into three Motion Homogeneity Categories (MHCs): 1) with homogenous motion, 2) with moderate motion, and 3) with complex motion. Then, the same MHC is assigned for all the CUs with in each CTU.

Since SHVC is an extension of HEVC, it inherits the CTU partitioning structure of HEVC. In SHVC, each frame is divided into several CTUs. Then during the CTU partitioning process, each CTU is divided into CUs of smaller sizes. Each CU (parent) is then divided into smaller CUs (child nodes) and this process is continued for D iterations. D indicates the maximum depth of the quad-tree. Note that each time the encoder splits the parent CU into four CUs. Therefore, the child CU is the part of parent CU in the lower level. It is expected to find high correlation between the mode (with the lowest RD cost) of the child CU and the parent CU. Therefore, in this study we decided to use the LRC mode of the parent CU as the third predictor feature. We call the parent CU as the parent predictor CU. Note that for the CUs located at quad-tree depth 0, the parent CU is not available.

In the case of SHVC, since there is a high correlation between the base layer and the enhancement layers, we use the LRC mode of the corresponding block in the BL as the fourth predictor feature. In our implementation for the quality scalability there are two ELs in addition to the BL. Therefore, for the current CU in the second EL (EL2), we add the LRC mode of the corresponding CU in the EL1 to the predictor feature list. We call the corresponding CU in the BL and the corresponding CU in the EL1 as the reference predictor CUs, hereafter.

Also, the modes of the already encoded neighboring CUs in the enhancement layer are valuable for predicting the mode of the current EL CU. In this regard, we add the LRC modes of the four neighboring CUs [29] in the current EL to the predictor feature list. We call these four CUs as the neighboring predictor CUs. Fig. 3.9 and Fig. 3.10 show the temporal, neighboring, and reference (previous layer/layers) predictor CUs for the to-be-coded EL1 CU and EL2 CU, respectively. Note that in addition to the predictor CUs showed in Fig. 3.9 and Fig. 3.10, we use the parent CU for the CUs located at quad-tree depth larger than zero.

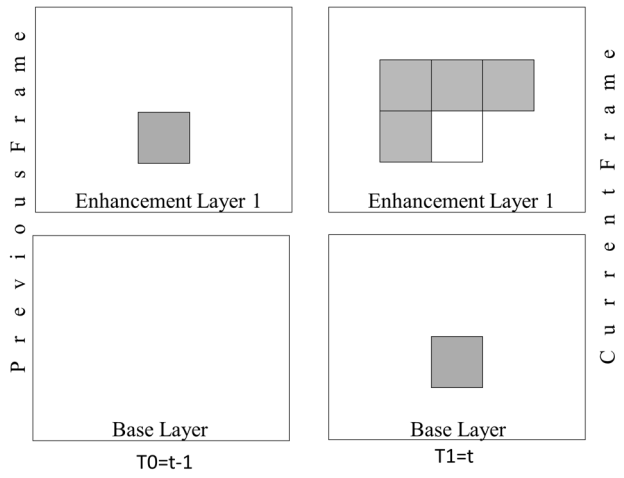

Figure 3.9 Current EL1 CU (white block) and its temporal, neighboring, and reference predictor CUs (Gray blocks).

The second step is the model generation or equivalently learning the posterior probability ($P(Y|X)$). The estimation of this posterior probability can be modeled as a supervised learning problem. In supervised learning a set of examples $D=\{(Y^1,X^1), (Y^2,X^2),\ldots, (Y^n,X^n)\}$ are used to build (train) the posterior probability distribution ($P(Y|X)$). We call this process as the model generation [105]. During the model generation process, the encoder encodes the BL using the unmodified SHVC [94]. The SHVC encoder, in the inter/intra prediction mode selection process, calculates the RD cost for each mode and the one with minimum RD cost is selected. Then, the ELs are also encoded using the conventional SHVC. In this process all of the available inter-intra modes are checked to find the lowest rate distortion cost. For each CU in the EL (EL1 or EL2), the information about the mode chosen by the encoder is stored. Based on this information, the probability of each mode in the current CU in EL (given the coding features of the predictor CUs) is updated. These conditional probabilities will then be used in the model exploitation process. In the model exploitation process, the information which is stored during encoding the BL and the previously encoded CU in the EL will be used to estimate a mode for the to-be-encoded CU in the EL. Therefore, the encoder does not check all the inter/intra prediction

modes. In rest of this Section, the equations which are required for the model generation process and model exploitation process will be presented.

As mentioned in the previous paragraphs, a probability can be assigned to each mode of the current CU at l^{th} EL and CTU depth layer d given the predictor features (i.e. P(mode of current CU at l^{th} EL and CTU depth layer d | LRC modes of temporal, neighboring, and the reference predictor CUs at CTU depth layer d, motion homogeny of the temporal predictor CU and the mode of parent CU at quad-tree depth layer *d-1*)). The posterior probability $P_{ld}(Y|X)$ (which determined the probability of the mode of the to-be-encoded CU at l^{th} EL and quad-tree depth layer d given the predictor features) can be calculated using the Bayes rule:

$$P_{ld}(Y|X) = \frac{P_{ld}(X|Y)P_{ld}(Y)}{P_{ld}(X)} = \frac{P_{ld}(X|Y)P_{ld}(Y)}{\sum_Y P_{ld}(X|Y)P_{ld}(Y)} \tag{3.17}$$

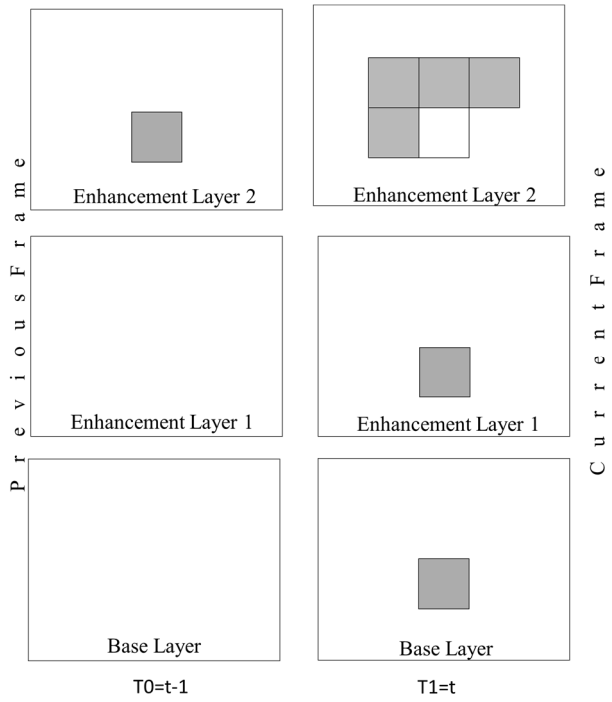

Figure 3.10 Current EL2 CU (white block) and its temporal, neighboring, and reference predictor CUs (Gray blocks).

To train a classifier each of $P_{ld}(X|Y_d)$ and $P_{ld}(Y)$ should be estimated. $P_{ld}(Y)$ is our prior belief of the mode of the to-be encoded CU. $P_{ld}(X|Y)$ is the class-conditional density which defines the distribution of data that is expected to be seen in each class. In our study $P_{ld}(X|Y)$ indicates the probability of the predictor features given each mode of the to-be-coded CU in the current EL (i.e. P(predictor features | mode of current CU at l^{th} EL and CTU depth layer d)).

Assuming that X and Y are discrete random variables, we should estimate a probability value for all possible combinations of X and Y. Suppose that there are M different values for different SHVC modes which would result in M different values for the random variable Y. The learning algorithm needs to estimate $M-1$ different parameters to estimate $P_{ld}(Y)$, since the probability should sum to one. On the other hand, X is a vector with N components. The first component of X (X_1) takes three possible discrete values (MHC classes). On the other hand, the other components of X (X_2 *to* X_N) have M possible discrete values. In this study, N is equal to the number of predictor features. Therefore, we have $3\times M^{N-1}$ different values for X. Estimating $P_{ld}(X|Y)$ requires learning of an exponential number of parameters i.e. $O(M^N)$, which is an intractable problem. Thereby, the key to use Bayes rule is to specify a suitable model for $P_{ld}(X|Y)$.

In our study to solve the above-mentioned intractable problem, Naive Bayes classifier [90] has been used. Naive Bayes classifier dramatically reduces the complexity of estimating $P_{ld}(X|Y)$ by making a conditional independence assumption. This learning algorithm assumes that the features of the predictor CUs are independent given Y. Taking into account the conditional independence assumption we have:

$$P_{ld}(X|Y) = P_{ld}(X_1, X_2, \dots, X_N|Y) = P_{ld}(X_1, X_2|Y) * \prod_{n=3}^{N} P_{ld}(X_n|Y) \quad (3.18)$$

Therefore,

$$P_{ld}(Y|X) = \frac{P_{ld}(X_1, X_2|Y) * \prod_{n=3}^{N} P_{ld}(X_n|Y) P_{ld}(Y)}{\sum_Y P_{ld}(X_1, X_2|Y) * \prod_{n=3}^{N} P_{ld}(X_n|Y) P_{ld}(Y)} \quad (3.19)$$

This simplifying assumption makes the representation of $P_{ld}(X|Y)$ simpler and reduces the number of parameters from an exponential term to just $O((N+3)\times M^2)$. To find $P_{ld}(Y|X)$, we need to have $P_{ld}(X|Y)$ and $P_{ld}(Y)$. These probabilities are computed during the model generation process. A very popular method to estimate these probabilities is using the MLE [89]. MLE estimate for $P_{ld}(Y)$ is:

$$P_{ld}(Y = k) = \frac{N_{ldk}}{T_{ld}} \quad (3.20)$$

where k indicates the possible values (modes) for Y. The element N_{ldk} is the number of observed instances of class k. That is, N_{ldk} indicates the number of times the modes of the CUs at l^{th} EL and CTU depth layer d is equal to k. T_{ld} denotes the total number of times the encoder determines the LRC modes for the CUs in the l^{th} EL and CTU depth layer d. Intuitively, $P_{ld}(Y=k)$ is the percentage of observing k as the mode of the current CU at l^{th} EL and CTU depth layer d. For the temporal predictor CU the estimate for $P_{ld}(X_1,X_2|Y)$ is as follows:

$$P_{ld}(X_1 = h, X_2 = q|Y = k) = \frac{N_{ldhqk}}{N_{ldk}} \quad (3.21)$$

where h indicates the possible values (MHC) for X_1. q shows possible values (modes) for X_2. The element N_{ldhqk} denotes the number of times $\{X_1=h,X_2=q\}$ has been observed in the instances of class k . Note that the $P_{ld}(X_1=h, ,X_2=q \ |Y=k)$ indicates the percentage of time the MHC of the temporal predictor CU is equal to h and its LRC mode is equal to q, while the mode of the current CU at the l^{th} EL and CTU depth layer d is equal to k. On the other hand the MLE estimate of $P_{ld}(X_n|Y)$for the reference and neighboring predictor CUs is as follows:

$$P_{ld}(X_n = m|Y = k) = \frac{N_{ldnmk}}{N_{ldk}} \quad \text{, for n=3,..,N} \quad (3.22)$$

where m indicates the possible values (modes) for X_n. The element N_{ldnmk} denotes the number of times $X_n=m$ has been observed in the instances of class k. That is, N_{ldnmk} indicates the number of times the modes of the n^{th} predictor is equal to m, while the mode of the current CU at the l^{th} EL and CTU depth layer d is equal to k. Intuitively, $P_{ld}(X_n=m \mid Y=k)$ indicates the percentage of observing m as the mode of the n^{th} predictor CU, while the mode of the current CU at the l^{th} EL and CTU depth layer d is equal to k.

In MLE estimation, there are some situations in which some states (i.e. $\{X=x_i, Y=y_i\}$) are not available in the training set. Therefore, after model generation, $P_{ld}(X=x_i|Y=y_i)=0$ which makes the Bayesian equation (3.19) equal to zero. In this case the classifier will have problem during the test process. This is an example of over-fitting to the training data [89]. So, in this study MAP [91] estimation is employed to resolve this problem. MAP estimate is a regularization of MLE which resolves the above-mentioned over fitting problem by incorporating a prior distribution over the parameter that we want to approximate. This prior distribution will be assigned using the prior knowledge which we have about those modes.

In order to facilitate MAP estimation, it is required to assign appropriate prior distribution for the parameter. The distribution of the $P_{ld}(Y|X)$ in this study is a categorical (multinomial) distribution, due to the fact that the number of modes which we have here is more than two. In fact categorical distribution is a distribution which describes the probabilities of the random event which have M (in this study M is equal to number of modes) outcomes and explains the probabilities separately. The solution to the MAP estimate for $P_{ld}(Y)$ is:

$$P_{ld}(Y = k) = \frac{N_{ldk}+\alpha_{ldk}}{T_{ld}+\sum_{j=1}^{M}\alpha_{ldj}}, \tag{3.23}$$

where α_{ldk} determines the strength of the prior assumptions relative to the observed data and M is equal to the number of different values which Y can take. For the temporal predictor, the MAP estimate for the $P_{ld}(X_1, X_2|Y)$ is as follows:

$$P_{ld}(X_1 = h, X_2 = q|Y = k) = \frac{N_{ldhqk}+\beta_{ldhqk}}{N_{ldk}+\sum_{i=1}^{3}\sum_{j=1}^{M}\beta_{ldijk}}, \quad (3.24)$$

where the element β_{ldhqk} determines the strength of the prior assumptions relative to the observed data. On the other hand the MAP estimate for $P_{ld}(X_n|Y)$ is as follows:

$$P_{ld}(X_n = m|Y = k) = \frac{N_{ldnmk}+\gamma_{ldnmk}}{N_{ldk}+\sum_{o=1}^{M}\gamma_{ldnok}}, \text{ n=3,..,N} \quad (3.25)$$

where the element γ_{ldnmk} determines the strength of the prior assumptions relative to the observed data and M is equal to the number of distinct values which X_n can take. To find the hyper parameters (α_{ldk}, β_{ldhqk}, and γ_{ldnmk}), we encode four representative video sequences selected from the MPEG dataset [92] for the HEVC call for proposals. These sequences include: PartyScene (832×480, 50fps), BQMall (832×480, 60fps), Racehorse (832×480, 30fps), and Vidyo3 (1280×720, 60 fps). These training video sequences are different from our test video sequences. In our study the random access main configuration [95] of SHVC is utilized [94]. Note that in this configuration, SAO and RDOQ are enabled [44]. The quantization parameters of the BL, EL1, EL2 were set to four different values (QPB, QPEL1, QPEL2)= {(26,22,18) ,{30,26,22) ,{34,30,26} ,{38,34,30}) [93]. To build the initial model (during the model generation), the training videos are encoded using unmodified SHVC and four QP sets. The coding information (LRC modes and MHC values) of all of the blocks (of different sizes) is stored in memory. Then, this information are utilized to compute the $P_{ld}(Y)$, $P_{ld}(X_1,X_2|Y)$, and $P_{ld}(X_n|Y)$ using (3.20), (3.21), and (3.22), respectively. Then we set $\alpha_{ldk}= P_{ld}(Y=k)$, $\beta_{ldhqk}= P_{ld}(X_1=h, X_2=q|Y=k)$, and $\gamma_{ldnmk}= P_{ld}(X_n=m|Y=k)$. The third step is to utilize the optimal Bayes decision rule [91] to classify a new X:

$$y_{mld} = \operatorname{argmax}_{y_{mld}} P_{ld}(Y|X) = \operatorname{argmax} P_{ld}(Y = y_{mld}) * P_d(X_1, X_2|Y = y_{lmd}) *$$

$$\prod_{n=3}^{N} P_d(X_n|Y = y_{mld}) \quad (3.26)$$

where y_{mld} is the m^{th} possible value of Y. The normalization part, i.e. $P_{ld}(X)$, of the posterior distribution has been omitted due to the fact that the denominator does not depend on y_{mld}. The resulting y_{mld} is the predicted mode for the current to be encoded CU.

3.2.3.2 Online-learning Based FMA

Unlike the method proposed in Subsection 3.2.2, in this study we implement our FMA based on online-learning. Our probabilistic model is fine-tuned during the course of encoding. If a new sample is observed, all the related probabilities are updated. Our FMA uses four probabilistic rules and four RD cost rules for mode prediction. By using these rules, this FMA is able to check a smaller number of modes to find the LRC modes for the EL CUs compared to the method proposed in Section 3.2.2. The main goal of our proposed FMA method is to gradually update the proposed probabilistic model and to utilize it to predict the LRC mode for the EL CUs. The mode-decision making process inside the SHVC encoder is content dependent. That is, once the content changes, the modes are expected to change too. In addition, modes also depend on the chosen encoding configuration which determines the available modes and other related parameters (e.g., search range, GOP, etc.). Moreover, modes are also dependent on the quantization parameters. In other words, the model and its efficiency will be highly dependent on the configuration and the quantization parameters that are used during training. To address this issue, we decided to implement our FMA using an online learning approach. In online learning, the main goal is coming up with an initial prediction model that is updated over time.

Similar to other machine learning approaches, the main target is predicting labels (modes with the lowest RD-cost in this case).

For the test video sequence, unmodified SHVC is utilized to encode the BL. As explained before, we want to implement our FMA based on online-learning. Our online-learning approach has two steps: model exploitation and model exploration. In the model exploitation, the probabilistic model is utilized to predict the probabilities of all the available inter/intra modes for the to-be-coded EL CUs. Then, among all the available modes only few modes are checked that are selected from the most probable modes. We call the modes that are suggested by our FMA during the model exploitation process as the mode candidates.

Model exploitation is done in the case that the mode candidates are more likely to be the LRC mode. However, in the case that the mode candidates are not reliable the model exploration process is utilized in which unmodified SHVC is utilized for the current EL CU and its coding information is utilized to accurately fine-tune (update) the model. Therefore, the model exploration improves the accuracy of the model. But, during this process the computational complexity is not reduced. As we increase the number of EL CUs that are encoded during the

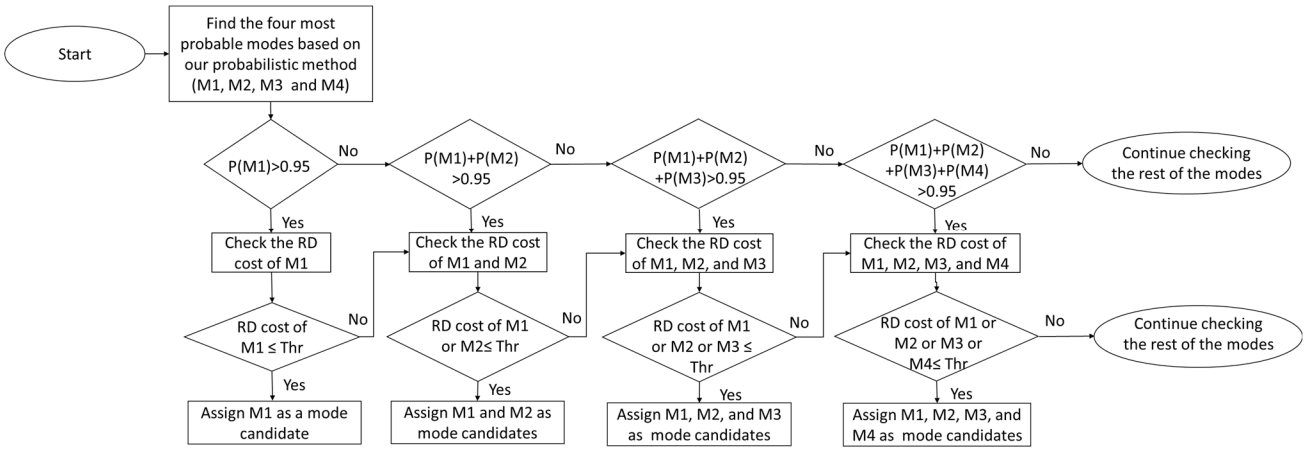

Figure 3.11 Block diagram of our proposed online learning based FMA for SHVC encoder.

model exploitation process, the complexity reduction performance is increased and the accuracy may be reduced. Therefore, we need to find when it is required to explore the model and when the model exploitation process can be utilized. In this regard, two conditions are considered; probability condition and RD cost threshold condition. The probability of the mode candidates should be very high (larger than 0.95), thus resulting in insignificant bite-rate increase. In this regard, we need to find the number of most probable modes that can be considered as the mode candidates. We have conducted several experiments on our training video sequences to find the appropriate number of mode candidates. Our studies have shown that by increasing the number of mode candidates from one to four, the complexity reduction performance of our online-learning based FMA improves significantly. However, our method's performance seems to reach a saturation point above four modes, leading into only slightly complexity reduction. For this reason, we decided to check only four most probable modes as the mode candidates.

Fig. 3.11 illustrates the block diagram of the proposed online-learning based FMA. Here, the coding information of all the available predictor CUs and equation (3.26) are utilized to predict the most probable modes. These modes are suggested as the mode candidates that are more likely to be the LRC mode. In the case that all the neighbouring predictor CUs are not available, the reference predictor CUs and the temporal predictor CU are utilized for prediction. We compare the probability of the suggested modes with 0.95. In Section 3.1.2, an additive model is suggested to predict the RD cost of to-be-coded block in the EL which uses the RD cost values of the four neighboring blocks in the EL, the corresponding block in the BL (BL block), and the four neighboring blocks of the BL block in the BL. We call the four neighboring blocks in the EL, the corresponding block in the BL (BL block), and the four neighboring blocks of the BL block in the BL as the RD cost predictor blocks. The additive model has four weighting

constants that are computed using linear least square in Section 3.1.2 based on the available RD cost predictor blocks. Finally, an approach is proposed in Section 3.1.2 that uses the predicted RD cost to estimate RD cost threshold (Thr) for early termination of mode search process of quality scalable HEVC. More precisely, by using this approach for each of the to-be-encoded CU in EL, an RD cost threshold is estimated. In order to increase the chance of selecting the right mode (mode with the lowest RD cost), we compare the RD cost values of the modes that are suggested for current CU with the RD cost threshold that is predicted for that CU. In this regard, during encoding process the mode information (mode with the lowest RD cost and its RD cost) of all the CU sizes (which are checked during the quad-tree CTU partitioning) of the BL and ELs is stored in order to be used for RD cost threshold estimation when the size of the RD cost predictor blocks and the to-be-encoded CU in the EL (EL1 or EL2) are not similar. Then, we use the RD cost information of the RD cost predictor CUs to estimate the RD cost threshold to predict the RD cost threshold for the current EL CU. Note that for RD cost estimation, we use the weighting constants suggested in Section 3.1.2. As can be seen in Fig. 3, first we check if the probability of the first most probable mode is more than 0.95 and its RD cost is not larger than the RD cost threshold, the FMA only checks the first most probable mode. Otherwise, if the cumulative probability of the first and the second most probable modes is more than 0.95, and the RD cost values of at least one of them is not greater than the RD cost threshold, the FMA only checks the first and the second most probable mode. If the first and the second most probable modes are not able to fulfil the probability condition or the RD cost threshold, the cumulative probability of the first, the second, and the third most probable modes is compared with 0.95. If those three most probable modes cannot fulfil the probability condition or they cannot fulfil the RD cost threshold, the cumulative probability of the four most probable modes

are compared with 0.95. If the mode candidates cannot fulfil the probability condition or the RD cost threshold, the FMA is not able to suggest the mode. In the case that the FMA fails to find the mode, the modified encoder checks all the available modes (model exploration process). Note that the RD cost threshold is not defined in the case that less than two neighbours are available (see Section 3.1.2). Therefore, in this case our FMA is not able to suggest the mode candidates. In all the cases that our FMA is not able to suggest the mode candidates, the modified encoder will check all the unchecked modes.

Finally, the probabilistic model (probabilities) will be updated using the LRC mode of current EL CU. More precisely, when the mode of the current CU at l^{th} EL and CTU depth layer d is equal to k we increment N_{ldk} and T_{ld} by one. In this case

if $\{X_1=h,X_2=q\}$, we increment N_{ldhqk} by one.

If $X_n=m$, we increment N_{ldnmk} by one (for n>2).

Note that the model is updated after both the model exploitation process and model exploration process. As can be seen in Fig. 3, the FMA may check one to four modes based on the four probabilistic rules and the four RD cost rules. Therefore, the complexity reduction based on the probabilistic model leads to significant computational complexity reduction for encoding CUs in EL.

3.3 Experimental Results and Discussions

In our experiment, in order to evaluate the performance of the proposed complexity reduction schemes, seven test sequences, which are suggested for the SHVC, from the dataset provided by MPEG are utilized [93]. Table 3.6 summarizes the specifications of this test dataset. All the methods presented in this Chapter are implemented to the SHVC reference software (SHM 6.1

[94]). In our implementation, we have two enhancement layers (EL1 and EL2) in addition to the BL for the quality scalability.

Same as the training process of Section 3.2.2 and Section 3.2.3, the random access main configuration [95] of SHVC is used for the test video sequences. The quantization parameters of the BL, EL1, EL2 were set to four different values (QP_B, QP_{EL1}, QP_{EL2})= {(26,22,18), (30,26,22), (34,30,26), (38,34,30)) [93].

The performances of our proposed complexity reduction methods are compared with that of the unmodified SHVC encoder in terms of execution time, and impact on bitrate and PSNR. For the quality scalability, we have examined six scenarios for the EL: 1) Early Merge Decision [28] (just for the EL), 2) Content adaptive complexity reduction scheme (CARCS) presented in Section 3.1, 3) Hybrid complexity reduction based on statistical studies presented in Section 3.2.1 4)Naive based fast mode assignment (NFMA) proposed in Section 3.2.2, 5) online learning based FMA proposed in Section 3.2.3, and 6) The combination of online learning based FMA and content adaptive complexity reduction scheme. In order to investigate the importance of the fine-tuning process (for the online learning based FMA), we also implement the proposed FMA without online-learning (to investigate the effect of disabling the online-learning approach). It means during the encoding process of the test video sequences, the probabilistic

Table 3.6 Test video dataset specifications.

Name	Resolution	Frame Rate (fps)	Number of frames
Traffic	2560×1600	30 Hz	150
PeopleOnStreet	2560×1600	30 Hz	150
Kimono	1920×1080	24 Hz	240
ParkScene	1920×1080	24 Hz	240
Cactus	1920×1080	50 Hz	500
BasketballDrive	1920×1080	50 Hz	500
BQTerrace	1920×1080	60 Hz	600

model is not updated.

The results of our experiment are reported in Table 3.7, Table 3.8, and Table 3.9. For each method the time reduction percentage compared to unmodified SHVC encoder as well as the impact on bitrate (Bjontegaard delta rate (BD-BR) [99]) and video quality in terms of PSNR (BD-PSNR [99]) are reported. Note that for each video sequence three time reduction percentage values are reported including time reduction percentage for EL1, time reduction percentage for EL2, and total time reduction percentage (BL+EL1+EL2). In our experiment a blade with an Intel Xeon X5650 6-core processor, running at 2.66GHz, and 8-GB RAM from the Bugaboo

Table 3.7 The impact of all the methods on bitrate and BD-PSNR for the test video sequences under random access main configuration.

Methods Tested in Our Study	BD-BR, BD-PSNR	Test Video Sequences							
		Traffic	PeopleOnStreet	Kimono	ParkScene	Cactus	BasketballDrive	BQTerrace	**Average**
Early merge descion (EMD)	EL1 BD-PSNR (dB)	-0.038	-0.020	-0.026	-0.075	-0.017	-0.042	-0.042	**-0.037**
	EL1 BD-BR	1.32%	0.46%	0.95%	2.39%	1.06%	2.23%	2.38%	**1.54%**
	EL2 BD-PSNR (dB)	-0.069	-0.035	-0.053	-0.079	-0.033	-0.036	-0.047	**-0.050**
	EL2 BD-BR	2.44%	0.73%	2.44%	2.65%	1.63%	1.66%	2.82%	**2.05%**
Content adaptive complexity reduction scheme (CARCS)	EL1 BD-PSNR (dB)	-0.031	-0.006	-0.055	-0.033	-0.002	-0.021	-0.0257	**-0.025**
	EL1 BD-BR	1.05%	0.02%	1.88%	1.10%	1.04%	1.44%	1.99%	**1.22%**
	EL2 BD-PSNR (dB)	-0.100	-0.061	-0.064	-0.090	-0.038	-0.041	-0.041	**-0.062**
	EL 2BD-BR	2.65%	1.32%	2.81%	3.05%	2.23%	2.53%	2.02%	**2.37%**
Naive Bayes Fast mode assignment (NFMA)	EL1 BD-PSNR (dB)	-0.002	-0.025	-0.039	-0.075	-0.017	-0.0423	-0.075	**-0.039**
	EL1 BD-BR	1.32%	0.07%	1.97%	2.39%	1.06%	2.23%	2.39%	**1.63%**
	EL2 BD-PSNR (dB)	-0.068	-0.09	-0.056	-0.0326	-0.0326	-0.036	-0.079	**-0.056**
	EL2 BD-BR	2.42%	1.95%	1.94%	2.65%	1.63%	1.63%	2.65%	**2.12%**
Hybrid complexity reduction	EL1 BD-PSNR (dB)	-0.075	-0.061	-0.030	-0.091	-0.051	-0.077	-0.039	**-0.061**
	EL1 BD-BR	1.88%	1.35%	1.06%	2.91%	2.17%	2.37%	2.94%	**2.10%**
	EL2 BD-PSNR (dB)	-0.015	-0.215	-0.060	-0.027	-0.110	-0.089	-0.050	**-0.081**
	EL2 BD-BR	4.18%	2.48%	2.76%	1.27%	3.65%	4.29%	2.70%	**3.05%**
Proposed FMA without online-learning	EL1 BD-PSNR (dB)	-0.008	-0.019	-0.041	-0.021	-0.016	-0.019	-0.031	**-0.022**
	EL1 BD-BR	1.22%	0.11%	1.65%	1.85%	1.39%	1.72%	1.89%	**1.40%**
	EL2 BD-PSNR (dB)	-0.043	-0.08	-0.049	-0.005	-0.029	-0.045	-0.025	**-0.039**
	EL2 BD-BR	2.37%	2.01%	2.11%	2.33%	2.15%	2.05%	2.13%	**2.16%**
Proposed online-learning based FMA	EL1 BD-PSNR (dB)	**-0.001**	**-0.015**	**-0.017**	**-0.015**	**-0.015**	**-0.01**	**-0.019**	**-0.013**
	EL1 BD-BR	**0.52%**	**0.08%**	**0.71%**	**0.97%**	**0.65%**	**0.82%**	**0.92%**	**0.67%**
	EL2 BD-PSNR (dB)	**-0.039**	**-0.004**	**-0.019**	**-0.009**	**-0.011**	**-0.023**	**-0.0359**	**-0.020**
	EL2 BD-BR	**1.11%**	**1.01%**	**1.21%**	**1.25%**	**0.99%**	**1.31%**	**1.01%**	**1.13%**
Proposed online-learning based FMA+CARCS	EL1 BD-PSNR (dB)	**-0.003**	**-0.016**	**-0.019**	**-0.016**	**-0.017**	**-0.012**	**-0.022**	**-0.015**
	EL1 BD-BR	**0.68%**	**0.13%**	**0.96%**	**1.14%**	**0.84%**	**1.05%**	**1.22%**	**0.86%**
	EL2 BD-PSNR (dB)	**-0.050**	**-0.010**	**-0.026**	**-0.018**	**-0.015**	**-0.027**	**-0.040**	**-0.027**
	EL2 BD-BR	**1.39%**	**1.17%**	**1.56%**	**1.68%**	**1.23%**	**1.57%**	**1.22%**	**1.40%**

Dell Xeon cluster from WestGrid (a high performance computing consortium in Western Canada).

As it is observed from Table 3.7 and Table 3.8, on average, the early merge method reduces the EL1 coding execution time by 55.06% for EL1 at the cost of 1.54% BD-BR increase for EL1, the EL2 coding execution time by 40.76% at the cost of 2.05% BD-BR increase for EL2, and the total execution time (BL+EL1+EL2) by 33.73% compared to the unmodified SHVC encoder. The CARCS method reduces EL1 coding execution time by 43.11% at the cost of 1.22% BD-BR increase for EL1, the EL2 coding execution time by 40.47% at the cost of 2.37% BD-BR increase for EL2, and the total execution time (BL+EL1+EL2) by 36.84% compared to the unmodified SHVC encoder. On average, the execution time reduction obtained by the NFMA method are 52.80% over SHVC at the cost of 1.63% BD-BR for EL1 and 50.66% over SHVC at the cost of 2.12% BD-BR for EL2 as shown in Table 3.7 and Table 3.8. The NFMA method decreases the total (BL+EL1+EL2) execution time on average by 36.84%. On average, our proposed online-learning based FMA reduces the time execution time by 64.50% for EL1 at cost of 0.67% bit-rate increase and by 62.85% for EL2 at the cost of 1.13% bit-rate increase (see Table 3.7). This method achieves the total (BL+EL1+EL2) time reduction percentage of 45.40%. We also observe that our proposed hybrid scheme achieves the coding time reduction percentages of 72.40% for EL1 and 63.99% for EL2, on average. This method decreases the total execution time on average by 48.49% compared to unmodified SHVC. The average bitrate degradation values for this method are 2.10% BD-BR for EL1 and 3.05% BD-BR for EL2, on average.

The EL1, EL2, and the total (BL+EL1+EL2) time reduction achieved by our proposed online-learning method are on average 9.44%, 22.09%, and 11.67% compared to the EMD

method [21] (see Table III), respectively. The time reduction achieved by our proposed online-learning based mode prediction method for the EL1, EL2, and BL+EL1+EL2 are on average 21.39%, 22.38%, and 15.42% compared to CARCS (see Table II), respectively. Compared to NFMA method (Section 3.2.2), the EL1, EL2, and the total time reduction performances of our proposed online-learning based FMA method are 11.70%, 12.19%, and 8.56% on average, respectively for the random access main configuration.

As we have observed in Table 3.7 and Table 3.8, the time reduction and compression (BD-BR and BD-PSNR) performances of the NFMA are lower than those of our proposed online-learning based FMA. The superiority of our online-learning based FMA is achieved by using the online-learning approach, flexible number of mode candidates, probability condition, the RD

Table 3.8 The impact of all the methods on execution time reduction (TR) for the test video sequences under random access main configuration.

Methods Tested in Our Study	Time Reduction (TR) %	Test Video Sequences							
		Traffic	PeopleOnStreet	Kimono	ParkScene	Cactus	BasketballDrive	BQTerrace	**Average**
Early merge descion (EMD)	EL1 TR	64.64%	42.71%	51.78%	62.75%	55.35%	54.20%	53.96%	**55.06%**
	EL2 TR	49.57%	25.16%	44.44%	49.84%	39.32%	38.42%	38.58%	**40.76%**
	Total TR	39.73%	23.40%	35.80%	39.78%	32.95%	32.09%	32.34%	**33.73%**
Content adaptive complexity reduction scheme (CARCS)	EL1 TR	38.03%	43.00%	46.90%	41.15%,	41.93%	49.04%	39.77%	**43.11%**
	EL2 TR	38.65%	42.11%	39.13%	40.68%	41.93%	41.34%	39.44%	**40.47%**
	Total TR	27.96%	29.50%	30.79%	29.39%	30.53%	32.34%	29.35%	**29.98%**
Naive Bayes Fast mode assignment (NFMA)	EL1 TR	44.69%	54.98%	56.54%	55.90%	49.79%	52.44%	55.25%	**52.80%**
	EL2 TR	45.00%	45.73%	53.05%	55.27%	49.92%	51.28%	54.39%	**50.66%**
	Total TR	31.49%	35.39%	38.47%	39.93%	35.96%	36.10%	40.55%	**36.84%**
Hybrid complexity reduction scheme based on statistical studies	EL1 TR	70.93%	66.08%	76.40%	70.48%	74.65%	73.84%	74.40%	**72.40%**
	EL2 TR	58.97%	61.92%	62.19%	61.58%	68.12%	67.85%	67.28%	**63.99%**
	Total TR	45.63%	44.78%	49.36%	47.11%	49.95%	50.35%	52.28%	**48.49%**
Proposed FMA without online-learning	EL1 TR	52.88%	62.50%	64.28%	63.85%	56.60%	59.60%	57.93%	**59.66%**
	EL2 TR	53.35%	55.25%	62.08%	62.35%	55.25%	57.23%	58.60%	**57.73%**
	Total TR	37.55%	41.25%	44.89%	48.48%	40.25%	41.49%	43.12%	**42.44%**
Proposed online-learning based FMA	EL1 TR	60.50%	65.00%	68.00%	68.75%	62.00%	65.50%	61.75%	**64.50%**
	EL2 TR	59.00%	61.50%	65.75%	66.50%	61.25%	63.33%	62.63%	**62.85%**
	Total TR	42.28%	43.27%	47.53%	48.48%	44.42%	45.77%	46.04%	**45.40%**
Proposed online-learning based FMA + CARCS	EL1 TR	69.21%	72.72%	75.45%	78.38%	71.60%	76.73%	70.85%	**73.56%**
	EL2 TR	67.21%	71.29%	72.60%	75.14%	70.16%	72.11%	71.00%	**71.36%**
	Total TR	50.91%	51.56%	50.15%	51.42%	52.65%	52.59%	51.90%	**51.60%**

cost threshold condition, LRC mode of parent CU, and coding information of the temporal predictor CU. The NFMA method encodes the first second of the test video using unmodified SHVC to fine-tune its probabilistic model. Then for the rest of the frames of the test video it always checks two mode candidates for the ELs. However, unlike NFMA which is not able to reduce the complexity of encoding the first frame of the test video, our online-learning based FMA is able to reduce the computational complexity of encoding the ELs of the first second of the test video sequence.

As can be seen in Table 3.7 and Table 3.8, when the fine-tuning process is disabled (no online-learning is involved), the proposed FMA reduces the EL1 execution time by 59.66% at the cost of 1.40% bit rate increase for the EL1, the EL2 execution time by 57.73% at the cost of 2.16% for the EL2, on average for the random access main configuration. The proposed FMA without online-learning approach achieves average total execution time reduction of 42.44%.

In order to investigate the effects of online-learning approach (fine-tuning process) in more details, we compute the percentage of EL CUs when 1) one mode is checked, 2) only two modes are checked, 3) three modes are checked, and 4) four modes are checked. In addition, we also compute the percentage of checking up to four modes, which shows the percentage of EL CUs for which our FMA is able to suggest mode/modes that can fulfil the probability and RD cost threshold conditions (Fig. 3.10). Therefore, a higher value of percentage of checking up to four modes corresponds to more complexity reduction. Percentage of checking up to four modes value is equal to the summation of all the cases listed above. Moreover, we also report the percentage of hit ratio that shows the percentage of EL CUs for which our online-learning based FMA chooses the optimal modes. The optimal mode of each CU is the LRC mode that is chosen by the unmodified SHVC's encoder for that CU. The hit ratio determines the compression

performance of our proposed. The percentage of checking a different number of mode candidates and percentage of hit ratio are reported in Table IV for each of the test video sequences for the online-learning based FMA and FMA without online-learning.

As can be seen in Table 3.9, by enabling the fine-tuning process the average percentages of checking up to four modes are increased from 94.20% for EL1 and 93.37% for EL2 to 97.14% for EL1 and 97.20% for EL2. The fine-tuning process also increases the average percentages of checking only one mode from 87.02% for EL1 and 78.73% for EL2 to 94.85% for EL1 and 93.78% for EL2. Therefore, the fine-tuning process reduces the percentages of checking more than one modes. It also improves the percentages of hit ratio of our FMA method from 94.61% for EL1 and 94.09% for EL2 to 97.56% for EL1 and 97.12% for EL2, on average.

Our results show the importance of the fine-tuning process and the reason of including it in our final approach. The online-learning approach increases the probabilities of the most observed modes. Therefore, percentage of checking up to four modes and percentage of checking only one mode are higher in the case of using online-learning based FMA compared to FMA without online-learning. Thus, the complexity reduction performance of online-learning based FMA is better than FMA without online-learning. In addition, since in the case of online-learning approach the model is updated using the LRC mode that is selected for each EL CU, the model is adjusted to the video content and quantization. Therefore, the hit ratio of the case of using online-learning approach is more than the case of disabling the online-learning approach. Thus, the compression (BD-BR and BD-PSNR) performance of online-learning based FMA is better than FMA without online-learning.

As we have observed, the results show the superiority of our hybrid method and our proposed online-learning based FMA over other methods in terms of complexity reduction

performance. However, using the online-learning approach results in better compression performance compared to all the methods tested in Section 3. In order to achieve higher performance, we use the combination of our online-learning based FMA and our content adaptive complexity reduction scheme. The combined method reduces the EL1 execution time by 73.56% at the cost of 0.86% bit rate increase for the EL1, the EL2 execution time by 71.36% at the cost of 1.40% for the EL2, on average. This method decreases the total execution time on average by 51.60% compared to unmodified SHVC (see Table 3.7 and Table 3.8).

Table 3.9 Percentage of checking a different number of mode candidates, mode prediction success, and mode prediction accuracy for the proposed FMA without online-learning and the online-learning based FMA.

Methods	Layer	Percentage of	Test Video Sequences							
			Traffic	PeopleOnStreet	Kimono	ParkScene	Cactus	BasketballDrive	BQTerrace	**Average**
Proposed FMA without online-learning	EL1	Checking only one mode	81.09%	87.57%	90.60%	88.22%	87.29%	90.80%	83.60%	**87.02%**
		Checking only two modes	6.17%	4.03%	1.51%	3.31%	3.83%	2.42%	3.04%	**3.47%**
		Checking only three modes	2.80%	1.89%	1.42%	1.41%	1.58%	1.88%	3.41%	**2.06%**
		Checking only four modes	2.83%	1.79%	1.25%	1.37%	1.27%	0.75%	2.25%	**1.64%**
		Checking upto four modes	92.89%	95.28%	94.77%	94.31%	93.96%	95.85%	92.30%	**94.20%**
		Hit ratio	93.98%	97.97%	94.91%	93.74%	92.82%	94.05%	94.78%	**94.61%**
	EL2	Checking only one mode	70.78%	78.77%	83.18%	80.41%	79.22%	81.63%	77.11%	**78.73%**
		Checking only two modes	10.95%	7.39%	6.37%	9.00%	8.35%	7.24%	8.92%	**8.32%**
		Checking only three modes	5.19%	5.37%	3.58%	3.04%	4.25%	3.59%	5.03%	**4.29%**
		Checking only four modes	2.48%	2.85%	1.13%	1.51%	1.74%	2.11%	2.38%	**2.03%**
		Checking upto four modes	89.40%	94.38%	94.26%	93.96%	93.56%	94.57%	93.44%	**93.37%**
		Hit ratio	93.43%	94.71%	95.09%	93.54%	93.85%	94.93%	93.05%	**94.09%**
Proposed online-learning based FMA	EL1	Checking only one mode	92.43%	92.85%	96.78%	96.00%	96.52%	95.79%	93.58%	**94.85%**
		Checking only two modes	1.21%	2.43%	0.70%	1.11%	1.10%	1.23%	1.97%	**1.39%**
		Checking only three modes	0.55%	0.80%	0.29%	0.49%	0.34%	0.66%	0.92%	**0.58%**
		Checking only four modes	0.20%	0.38%	0.12%	0.30%	0.21%	0.41%	0.57%	**0.31%**
		Checking upto four modes	94.40%	96.46%	97.89%	97.89%	98.17%	98.09%	97.05%	**97.14%**
		Hit ratio	97.75%	98.41%	98.12%	97.02%	96.74%	96.93%	97.95%	**97.56%**
	EL2	Checking only one mode	85.59%	93.38%	96.20%	95.46%	96.14%	95.14%	94.55%	**93.78%**
		Checking only two modes	8.09%	2.35%	0.93%	1.65%	1.21%	1.44%	1.11%	**2.40%**
		Checking only three modes	0.70%	0.87%	0.57%	0.48%	0.54%	0.51%	0.94%	**0.66%**
		Checking only four modes	0.21%	0.44%	0.22%	0.29%	0.31%	0.50%	0.56%	**0.36%**
		Checking upto four modes	94.60%	97.05%	97.93%	97.89%	98.20%	97.60%	97.16%	**97.20%**
		Hit ratio	96.98%	97.44%	97.39%	96.63%	97.22%	97.16%	97.01%	**97.12%**

In summary, the results show the superiority of the combination of our online-learning based FMA and our content adaptive complexity reduction scheme over other methods tested in this Section.

3.4 Conclusions

This Chapter focuses on developing a complexity reduction schemes for the quality scalable extension of HEVC (SHVC) encoder. In this regard, first we proposed a content adaptive complexity reduction scheme for SNR/Quality scalable EVC. In our scheme the RD cost and motion information of the base layer is utilized to facilitate the inter prediction and intra prediction mode selection process in the enhancement layers by avoiding redundant computations. Performance evaluations show that our approach results in significant total (BL+EL1+EL2) SHVC coding complexity reduction (by 29.98% on average) at the cost of 1.54% bitrate increase for EL1 and 2.05% bitrate increase for EL2.

Moreover, we propose three mode prediction schemes. For the first scheme, we design two mode prediction approaches based on statistical studies, namely quad-tree based mode prediction approach and a reference layer mode-information based mode prediction approach. We combined a quad-tree based mode prediction method and a reference layer mode-information based mode prediction method to a hybrid complexity reduction scheme. The performance was tested over a representative set of video sequences and compared with the unmodified version of the SHVC encoder. These evaluations showed that this approach outperforms the other schemes presented in this Chapter (in terms of complexity reduction), reducing the encoding execution time by 48.49% on average at the cost 2.10% bitrate increase for EL1 and 3.05% bitrate increase for EL2.

Afterwards, in order to improve the compression efficiency of the hybrid complexity reduction method, we propose a fast mode-assigning (FMA) method based on Naive Bayesian classifier (the second mode prediction scheme) for reducing the complexity of quality scalable HEVC. This method uses the probabilistic model that is created using the training data. For the test video, the probabilistic model is fine-tuned using the coding information of early frames of the test video. For the rest of the videos, the method uses the mode information of four neighboring blocks in the EL and the co-located block in the BL to predict the mode of current EL block. The results show that our method decreases the total execution time of SHVC's encoder by 36.84% on average at the cost of 1.63% bitrate increase for EL1 and 2.12% bitrate increase for EL2. Using the fine-tuning processing and the Bayesian approach results in better compression performance compared to hybrid complexity reduction scheme.

Then, in order to improve the performance of the Naive Bayesian based FMA method, we introduce an online learning based FMA method to predict the mode with the lowest rate distortion cost for the coding units (CUs) in the enhancement layers. The online learning based FMA builds a probabilistic model that uses the mode information of the parent CU in the quad-tree structure of a coding tree unit to predict the probabilities of all the available inter/intra modes of current CU in the enhancement layer (EL). In addition, the FMA uses the mode information and motion homogeneity of already encoded CUs in the base layer and EL to build the probabilistic model. Finally, we propose an online-learning based FMA that uses our probabilistic model to predict the mode with the lowest rate distortion cost for the current CU in the EL. During the encoding process, online-learning is used to fine-tune the probabilistic model, using the mode information (feedback) it receives from the encoder. Performance evaluations show that our online-learning based FMA outperforms all the method studies in this Chapter

(except the hybrid complexity reduction method proposed based on statistical studies) by significantly reducing the execution time of the encoder (by 45.40% for the total encoding time for the random access main configuration, on average) at the cost of 0.67% bitrate increase for EL1 and 1.13% bitrate increase for EL2. Our experiments show that compared to all the methods presented in this Chapter, the online-leaning based FMA has the best compression performance which is achieved because of using the online-learning approach. In order to achieve higher complexity reduction performance, we use the combination of our online-learning based FMA with our content adaptive complexity reduction scheme which achieves the total time reduction of 51.60% on average at the cost of 0.86% bitrate increase for EL1 and 1.40% bitrate increase for EL2.

4. Complexity Reduction Scheme for 3D-HEVC

The past two Chapters have dealt with spatial and quality extensions of HEVC. In this Chapter we move to 3D extension of HEVC.

As stated earlier in Chapter 1, 3D-HEVC is a new emerging video compression standard for multiview video applications. This standard utilizes advanced interview prediction characteristics in addition to the prediction features of the HEVC standard for efficient encoding of multiview video content. While using combined features improve the compression performance by utilizing the correlation between the views captured from slightly different angles of the same scene, they also increase coding complexity. The focus of this Chapter is on developing an efficient complexity reduction schemes for the dependent texture views (DV_ts) of 3D-HEVC, with the intention to facilitate the adoption of this standard, especially for real-time applications. In this regard, we propose three complexity reduction schemes. First, we propose a content adaptive complexity reduction scheme. Our scheme uses an adaptive early-termination inter and intra prediction mode search that reduces the 3D-HEVC coding complexity by utilizing the correlations between views. This is done by utilizing specific coding information of one view such as motion homogeneity, prediction modes, and the RD cost as well as the disparity between that view and other views. In addition, in order to improve the efficiency of our content adaptive complexity reduction scheme, a low complexity mode decision approach is proposed which uses the mode information of four neighboring DV CUs and corresponding CU in the BV for mode prediction. The method uses the training video to generate its initial probabilistic model. For the test video, the first second of the test video is utilized to fine-tune the model. Then, for the rest of the videos Bayesian classifier is used for mode prediction.

Finally, to achieve higher complexity reduction performance, we propose an online-learning hybrid complexity reduction for the DV_ts of 3D-HEVC. The proposed scheme uses two probabilistic models to predict the mode with the lowest RD cost for the to-be-coded block in current (DV_t). The first probabilistic model utilizes the mode of the root blocks to predict the mode with the lowest RD cost for their children DV_t blocks. Our first probabilistic model considers if the parent blocks are all-zero blocks (AZBs) to predict the mode of current DV_t block. Similar to our low complexity mode decision approach, our second probabilistic model utilizes a Naive Bayes classifier that uses the mode information of the corresponding block in the base texture view and the four neighbouring blocks of the current DV_t to predict the mode of to-be-encoded block in the current DV_t. However, our second probabilistic model uses more coding information (mode of the corresponding block in the previous frame, AZBs and motion homogeneity) for mode prediction compared to our low complexity mode decision approach. Then, we propose a hybrid complexity reduction scheme, which utilizes the two probabilistic models, motion information of the base texture view, and the rate distortion cost of the already encoded blocks in the base and dependent texture views. Unlike the method proposed by other researchers [73]–[85] which do not use online-learning, our proposed scheme uses gradually fine-tuned probabilistic modeling based on content and the quantization parameters. Our online-learning based complexity reduction scheme finds the mode with the lowest RD cost for CUs in current DV_t by checking a smaller number of modes compared to the content adaptive complexity reduction scheme and the low complexity mode decision approach. Unlike the low complexity mode decision approach which uses unmodified 3D-HEVC encoder for the first second of each scene, the proposed scheme is able to start decreasing the complexity much earlier, depending on the video content. The performance of our proposed schemes is tested for

the case with two views (i.e. base view + a dependent view). The evaluations confirm that our proposed online-learning based hybrid complexity reduction scheme reduces the 3D-HEVC codec complexity significantly compared to the unmodified 3D-HEVC encoder while maintaining the overall video quality.

Our content adaptive complexity reduction scheme is presented in Section 4.1. Section 4.2 presents our low complexity mode decision approach. Our proposed online-learning based hybrid complexity reduction scheme is described in detail in Section 4.3. Experimental results are presented in Section 4.4. Finally, conclusions are given in Section 4.5.

4.1 Content Adaptive Complexity Reduction Scheme for 3D-HEVC

For 3D-HEVC in each time instant we have one independent view and one or more dependent views and their corresponding depth information which are encoded. On the decoder side, the display decodes the independent view. Then, if it has enough capacity it decodes one or more dependent views. As these views are captured from one scene, there are correlations among them. By utilizing this information we can predict the appropriate search range and RD cost of the to-be-encoded CU in the to-be-encoded view.

4.1.1 Adaptive Search Range Adjustment

As explained before, one of the most computationally expensive tasks in video coding is motion search in inter prediction. The larger the search range is the more expensive the computational cost becomes. On the other hand, choosing a very small search-range may reduce the compression performance due to poor matching results. By selecting an optimal motion

search range one may achieve to reduce the complexity without hampering the compression performance.

In the case of multiview coding, since the views are captured from a common scene, there is a correlation between the motion vectors of the views. In our study we utilize this correlation to select the proper motion search range for the dependent views based on the motion information of the base view. To do this, we first classify the CTUs within each frame in the base view to different categories in terms of search range as follows [71]:

$$SR_{CTU} = \begin{cases} round\left(\frac{SR}{64}\right) & if\ more\ than\ 85\%\ of\ MVs\ have\ \ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{64} \\ round\left(\frac{SR}{32}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{32} \\ round\left(\frac{SR}{16}\right) & else\ \ if\ more\ than\ 85\%\ of\ MVs\ have\ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{16} \\ round\left(\frac{SR}{8}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{8} \\ round\left(\frac{SR}{4}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ \ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{4} \\ round\left(\frac{SR}{2}\right) & else\ if\ more\ than\ 85\%\ of\ MVs\ have\ amplitude\ of\ less\ or\ equal\ to\ \frac{SR}{2} \\ SR & Otherwise \end{cases} \quad (4.1)$$

where *SR* indicates the search range defined in the configuration file of the codec, MVs represent the motion vectors of the CTU in the base view , and SR_{CTU} is the adjusted search range of the corresponding CTU in the dependent view (see Fig. 4.1). In the proposed approach, for coding the current CTU in the dependent view, the corresponding CTU in the base view is found by using the disparity information between the dependent view and the base view (which is derived from depth information). Note that depending on the category that the corresponding CTU in the base view belongs to, the search range of the current CTU in the dependent view is adjusted and as a result all the CUs within that CTU have the same adjusted motion search range setting. As it

can be observed from (4.1), the search range in the dependent views can become quite small, depending on the category the matching CTU in the base view belongs to. Taking into account that there might be several CUs (up to 64) within a CTU, this scheme will significantly reduce the computational cost of 3D-HEVC.

4.1.2 Early Termination Mode Search

During the 3D-HEVC inter prediction process, the encoder tests all three inter prediction modes (similar to HEVC-based encoder). The inter prediction modes include skip, merge, and explicit motion vector encoding. The encoder first checks for the skip and merge modes, which are computationally less expensive, compared to the explicit motion vector encoding process. Our objective here is to implement an early termination (ET) mode-search, so that the encoder does not need to go through all the modes, thus significantly reducing the computational complexity.

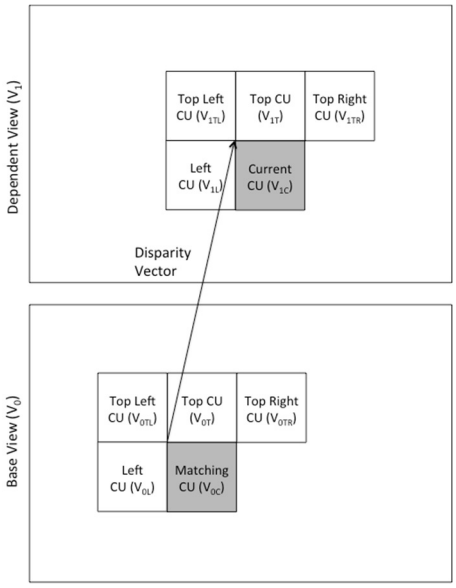

Figure 4.1 Current CU and its four spatial neighbors of base view and the current view.

The encoder in the inter/intra prediction mode selection process calculates the RD cost for each mode and the one with minimum RD cost is selected. In the mode search process, if the RD cost of the current to-be-coded CU in a dependent view is predicted from the already coded CUs, once the RD cost of a mode is close or equal to the predicted RD cost, the mode search can be terminated. This will significantly reduce the computational complexity. In our approach, to find a prediction for the RD cost of the current CU in the dependent view, we utilize the RD cost of the already coded adjacent CUs in the dependent view and that of their matching CUs in the base view. Fig. 4.1 shows an example of the arrangement of the CUs whose information is utilized to predict the RD cost of the to-be-coded CU in the dependent view.

Inspired by [46], we assume that there is an additive model between the RD cost of the CUs in the dependent view (V_1) and their corresponding CUs in the base view (V_0) as follows [71]:

$$RDcost\ V_{1c_{predict}} = \left(\alpha_1 \frac{RDcostV_{1T}}{RDcostV_{0T}} + \alpha_2 \frac{RDcostV_{1L}}{RDcostV_{0L}} + \alpha_3 \frac{RDcostV_{1TL}}{RDcostV_{0TL}} + \alpha_4 \frac{RDcostV_{1TR}}{RDcostV_{0TR}}\right) * RDcostV_{0c} \quad (4.2)$$

where $RdcostV_{1Cpredict}$ is the predicted RD cost of the current CU in the dependent view (V_1), $RdcostV_{0c}$ is the RD cost of the matching CU in the base view (V_0), $RdcostV_{1T}$, $RdcostV_{1L}$, $RdcostV_{1L}$ and $RdcostV_{1R}$ denote the RD cost of the four spatial neighbors of the current CU (see Fig. 4.1 for the arrangement of CUs), $RdcostV_{0T}$, $RdcostV_{0L}$, $RdcostV_{0TL}$and $RdcostV_{0TR}$ are the RD cost values of the matching CUs in the base view, and α_1, α_2 and α_3 are weighting constants. We compute these weighting constants in the following Subsection.

Once the predicted RD cost for the current to-be-coded CU is available, we define a threshold for early termination of the mode search in the dependent view as follows:

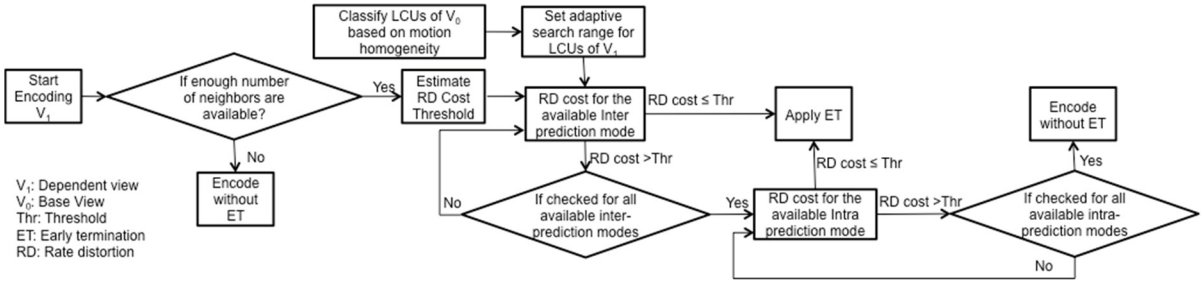

Figure 4.2 The block diagram of our 3D-HEVC encoder complexity reduction scheme.

$$Thr = \min(RdcostV_{1T}, RdcostV_{1L}, RdcostV_{1TL}, RdcostV_{1TR}, RdcostV_{1cpredict}) \quad (4.3)$$

By using this threshold, the encoder instead of testing all the modes, it terminates the mode search if the RD cost of a mode is less than the threshold, and selects that mode as the best one. Otherwise, it continues testing other modes until this criterion is met. Note that this scheme is applied to the CUs with at least two already-coded adjacent CUs.

In the case where the size of the matching CUs in the base view is not similar to the ones in the dependent view, the RD cost of the matching CU in the base view is normalized to its size and the RD cost used in our calculation is updated as follows:

$$RDcost_n = D\frac{w \times h}{W \times H} + \lambda * B \quad (4.4)$$

where W and H are respectively the width and height of the matching CU in the base view, w and h are the width and height of the current CU in the dependent view, D, λ and B are the distortion part, the Lagrangian constant value and the bit-cost of the matching CU in the base view respectively, and $Rdcost_n$ is the RD cost value to be used in finding the predicted RD cost and the threshold.

The threshold defined in (4.3) is also used in the intra prediction process of the dependent view to further reduce the complexity of the encoder. Fig. 4.2 provides a block diagram of our proposed complexity reduction scheme.

4.1.3 Determining the Weighting Constants

In order to find the proper weighting constants in equation (4.2), similar to Section 3.1.2.1 (of Chapter 3) the Linear Least Square method is used. Our objective is to minimize the difference between the predicted RD cost and the real RD cost of the best mode (without using ET) for the current to-be-coded CU in the dependent view. Our objective is formulated as follows:

$$\alpha_i = argmin_{\alpha_i} |(S - S')^2|, i = 0,1,2,3 \tag{4.5}$$

where S is a matrix that contains the real RD cost values of the best modes selected by 3D-HEVC for the current CU in the dependent view V_1 ($RDcostV_{1C}$) divided by $RdcostV_{0c}$, S' denotes a matrix which contains the predicted RD cost of the current CU ($RdcostV_{1Cpredict}$) divided by $RdcostV_{0C}$. We can re-write S' as follows:

$$S' = QA = \begin{bmatrix} q_{11}, q_{12}, q_{13}, q_{14} \\ q_{21}, q_{22}, q_{23}, q_{24} \\ . \quad . \quad . \quad . \\ . \quad . \quad . \quad . \\ q_{n1}, q_{n2}, q_{n3}, q_{n4} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \tag{4.6}$$

where

$$q_{i1} = \frac{RDcostV_{1Ti}}{RDcostV0_{Ti}}, \; q_{i2} = \frac{RDcostV_{1Li}}{RDcostV_{0Li}}, \quad q_{i3} = \frac{RDcostV_{1TLi}}{RDcostV_{0TLi}}, \; q_{i4} = \frac{RDcostV_{1TRi}}{RDcostV_{0TRi}} \qquad i=1,2,3,\ldots,n \tag{4.7}$$

where i indicates the index of the training data. From (4.6) the weighting constants are calculated as follows:

$$A=(Q^TQ)^{-1}Q^TS \tag{4.8}$$

We use a training dataset (four representative video sequences) to calculate the weighting constants. We code the video streams, record the real RD cost values, calculate the predicted RD cost based on the equation (4.2), and find the weighting constants based on the equation (4.8). In

case that all four spatial neighbors (T, L, TL and TR) are available (see Fig. 4.1), the estimated weighting constants are as follows: [α_1, α_2, α_3, α_4] = [0.34, 0.33, 0.16, 0.17]. When $RdcostV_{1TR}$ is not available, [α_1, $\alpha 2$, α_3, α_4] = [0.435, 0.42, 0.145, 0]. If $RdcostV_{1TL}$ is not available – which means that the $RdcostV_{1L}$ is not available either –we use two upper neighbors to predict the RD cost, and the weighting constants are [α_1, α_2, α_3, α_4] = [0.51, 0, 0, 0.49]. The weighting constants of the top and left neighboring CUs when available are larger than the others, denoting that they are more correlated with the current CU [71].

4.2 Low Complexity Mode Decision Approach for 3D-HEVC

The main objective of our study is to decrease the computational complexity of the 3D-HEVC encoder by utilizing the correlation between the base view (BV) and the dependent views (DVs). During the inter/intra prediction process, 3D-HEVC computes the RD cost for all of the available modes (based on the size of the to-be-encoded CU). Then, the encoder selects the mode that has the lowest RD cost. In our study, we propose a fast mode assigning (FMA) technique, which uses the mode information of the CUs in BV as well as the mode information of the already-encoded neighboring CUs in DVs to predict the mode of the to-be-encoded CU in each DV. This approach enables the encoder to avoid the extensive computational cost involved in the mode search process.

In our method for predicting the inter/intra prediction mode of the to-be-encoded CU in a DV, the mode information of the neighboring (top left, top, top right, and left) CUs that are already coded as well as the mode information of the corresponding CU in the BV are used. These CUs are called predictor CUs hereafter. Fig. 2 shows an example of a current to-be-coded CU in the n^{th} view and the predictor CUs, i.e., its four spatial neighbors and its corresponding

CU in the BV. Note that the neighboring CUs in the dependent view are similar to the candidates that 3D-HEVC chooses for the inter prediction merge mode.

Our goal here is to approximate a function whose input is the mode information of the predictor CUs and its output is the predicted mode of the current CU. In other words, we would like to estimate the posterior probability of the current CU's mode, given the mode information of the predictor CUs. This problem can be modeled as a supervised learning problem with training and testing processes. To formulate the problem, assume Y is the random variable corresponding to the probability of possible modes for the current to-be-coded CU in a DV, and X is a random vector corresponding to the probabilities of the modes of predictor CUs. If there are M different 3D-HEVC inter and intra modes, the random variable Y has M different values representing the probability of each mode. If there are L predictor CUs, then the length of vector X will be equal to L and each of its components can take M possible values which represent the probability. This results in M^L-1 different possible probability values for the random vector X. The term -1 comes from the fact that the probability values should sum to one. The probability of each mode of the current CU in DV given the probability of the modes of the predictor CUs, i.e., the posterior probability $P(Y|X)$, is calculated using the Bayes rule as follows:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{4.9}$$

where $P(Y)$ is the prior probability of the mode of the to-be encoded CU, $P(X|Y)$ is the class-conditional density, which defines the distribution probability of observing a combination of modes for predictor CUs given the probability of the mode of current CU.

To find $P(Y|X)$, the learning algorithms needs to estimate $P(Y)$ and $P(X|Y)$. The former requires estimating M-1values (variable Y can have M different values), and the later requires learning of an exponential number of parameters, which is an intractable problem [90]. In order

to estimate *P(X|Y)*,we use the Naive Bayes classifier [89]. The Naive Bayes classifier dramatically reduces the complexity of estimating *P(X|Y)* by making a conditional independence assumption. This learning algorithm assumes that different components of the *X* vector are independent with respect to a given *Y*. Taking into account the conditional independence assumption we have:

$$P(X|Y) = P(X_1, X_2, \dots, X_L|Y) = \prod_{l=1}^{L} P(X_l|Y) \tag{4.10}$$

Therefore,

$$P(Y|X) = \frac{\prod_{l=1}^{L} P(X_l|Y)\, P(Y)}{P(X)} \tag{4.11}$$

According to the optimal Bayes decision rule [89], the mode of the posterior probability distribution (predicted mode of the current CU in DV) is the mode, which has the largest probability among all the modes. Therefore, for classifying a new *X*, the following formula can be used:

$$\mathrm{y_m} = \underset{y_m}{\operatorname{argmax}}\, P(Y = y_m) \prod_{l=1}^{L} P(X_l|Y = y_m) \tag{4.12}$$

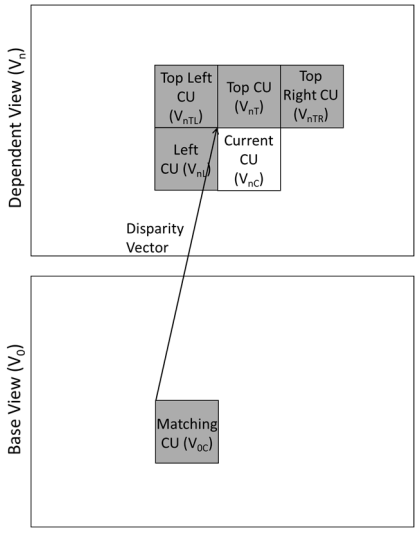

Figure 4.3 Current CU and its four spatial neighbors in base view and current view.

where y_m is the m^{th} possible value of Y. Note that $P(X)$ is the normalization factor in (4.11), thus it has been omitted in calculation of y_m (i.e., the value of y_m is independent of $P(X)$). The value of y_m represents the predicted mode for the to-be-encoded CU in DV.

To find the optimal value of y_m in (4.12), we need to have $P(X|Y)$ and $P(Y)$. These probabilities need to be computed during the training process. A very popular method to estimate these probabilities is the Maximum Likelihood Estimation (MLE) [89]. A major drawback of MLE is that when MLE is used for estimating the probabilities, there are some situations in which we have not seen some states (modes) in the training set. To resolve this problem, we employ Maximum a Posteriori (MAP) estimation [89], [106]. MAP estimation incorporates a prior distribution function over $P(X|Y)$ and $P(Y)$. In order to use MAP estimation, it is required to assign appropriate conjugate prior distribution for the parameters. The solution to the MAP estimation for $P(Y)$ is as follows:

$$P(Y = y_k) = \frac{N_k + \alpha_k}{Total\ number\ of\ tries + \sum_{k=1}^{M} \alpha_k} \tag{4.13}$$

where α_k determines the strength of the prior assumptions relative to the observed data, M is equal to the number of different values which Y can take, and N_k indicates the number of times the modes of the current CU is equal to y_k. Similarly the MAP estimation for $P(X|Y)$ is as follows:

$$P(X_l = x_{lm} | Y = y_k) = \frac{N_{lmk} + \alpha_{lm}}{Total\ number\ of\ tries + \sum_{m=1}^{M} \alpha_{lm}} \tag{4.14}$$

where α_{lm} determines the strength of the prior assumptions relative to the observed data, M is equal to the number of distinct values which X_l can take, and N_{lmk} indicates the number of times that the mode of the l^{th} predictor is equal to x_{lm}, given the mode of the to-be-encoded CU in DV is equal to y_k. To find the hyper parameters α_k and α_{lm}, which constitute the initial model, three

representative video sequences (see Section 4.4) are used in our approach. These video sequences are excluded from the video sets used to test our approach.

In order to improve the efficiency of the initial model and also make sure that it works for all possible encoding configurations, we fine-tune it by using the first few frames of each scene [106]. In our Naive Bayes FMA (NB-FMA) implementation (see Fig. 4.4), the first second of the video (i.e., 25 frames for 25fps format) is coded using a conventional 3D-HEVC encoder. Note that the number of frames is based on our empirical tests. The mode information of these frames is used for fine-tuning the model. During the training/fine-tuning process, the BV and DVs are encoded using the original 3D-HEVC encoder. For each CU in the DV, the information about the chosen mode is stored and the probability values are updated as the coding process continues. Based on this information, the probability of each mode (for a to-be-coded CU in DV) given the predictor CUs' mode is calculated. The 3D-HEVC modes (inter and intra modes) are labeled by discrete numbers and each mode is considered as a class.

During the training process, the probability values are updated as the coding process continues. In the testing process, first the BV is encoded, and then the encoder encodes the DVs. Unlike the training process, the encoder does not check all of the inter and intra prediction

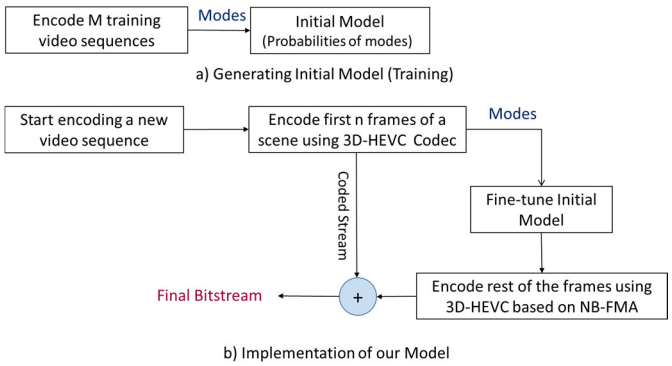

Figure 4.4 Block diagram of the proposed method.

modes. Instead, the modes of the predictor CUs are used for predicting the mode of the current CU. In this study, the three mode candidates with the highest probability among all the available modes are chosen, and the encoder calculates the RD cost for these three candidates and chooses the one with the smallest RD cost. If scene change occurs, the training process is repeated to update the probabilities of the model.

4.3 Bayesian Based Mode Prediction Method for 3D-HEVC

The objective of our work is to reduce the complexity of the 3D-HEVC codec by utilizing the coding information of the base texture view (BV_t) and the dependent texture views to minimize the redundant computations involved in coding the dependent texture views. In summary, first we attempt to decrease the complexity of the mode search process of the dependent texture view (DV_t) CUs using two different approaches, namely complexity reduction based on the quad-tree parenthood model, and complexity reduction using the neighborhood model. The quad-tree parenthood approach is based on a probabilistic model that uses correlations between different CUs of the same quad-tree structure to predict the probabilities of inter/intra modes of to-be-encoded CU in current DV_t. The second mode-search approach is a probabilistic model that utilizes the inter/intra modes and motion homogeneity of already encoded blocks in the BV_t and the current DV_t to predict the probability of inter/intra mode with the lowest RD cost (called LRC mode hereafter) for the to-be-encoded block in the current DV_t. The second probabilistic model also considers if the already encoded CUs are all-zero blocks. Unlike the method presented in Section 4.2, these two probabilistic models are fine-tuned during the encoding process.

In addition to reducing the complexity of the mode search process of 3D-HEVC, we also attempt to decrease the complexity of inter prediction using the adaptive search range adjustment proposed in Section 4.1.1.

Finally, we suggest a hybrid scheme that consists of a suitable combination of the above-mentioned approaches that results in the best possible performance. The following Subsections elaborate on our proposed scheme.

4.3.1 Mode Search Reduction Using Quad-tree Parenthood Model for Dependent Texture Views

4.3.1.1 Quad-tree Parenthood Model for Dependent Texture Views

The motion estimation process of 3D-HEVC includes two fast modes, the Skip mode and the Merge mode, as well as several inter and intra prediction modes, making it the most time-consuming part of the encoding procedure. To decrease the number of mode searching steps used by 3D-HEVC in dependent texture views, we need to design a scheme that avoids checking every possible option in order to identify the mode with the lowest RD-cost. In this Section, first we build a probabilistic model based on the coding information of the CUs belonging to the same quad-tree structure within a CTU in current DV_t to predict the probabilities of each available inter/intra prediction mode for the to-be-encoded blocks in current DV_t. Then, a FMA is

Table 4.1 Different CU sizes and their corresponding quad-tree depths (when the largest depth is equal to three).

Depth	0	1	2	3
CU size	A×A	A/2×A/2	A/4×A/4	A/8×A/8

proposed which uses this model to predict the mode that is more likely to be the LRC mode for the current DV_t CU.

Since 3D-HEVC is an extension of HEVC, it inherits the quad-tree structure of HEVC. During the encoding process in 3D-HEVC, each frame is divided into several square blocks of the same size (64x64) known as CTUs. In the first depth layer of the quad-tree structure, each 64x64 CTU can be divided into four 32x32 CUs. Here we consider the CTU to be the parent of all the CUs in the first depth layer. In the second depth layer of the quad-tree, each CU of the first quad-tree depth layer (called parent) can be split into four CUs (called children). Note that a CTU is the ancestor CU for all of the CUs in the second depth layer. Accordingly, a CU in the second depth layer of the quad-tree can be split into four CUs in the third depth layer. Fig. 3.4 in Chapter 3 illustrates a CTU and its three-layer quad-tree structure. The size of CUs at different quad-tree depth layers are presented in Table 4.1. Note that in Table 4.1, *A* is equal to the width/height of the CTU.

As seen in Fig. 3.3 of Chapter 3, the CUs at each quad-tree depth level are part of the parent CU in the lower level. Therefore, it is expected to find high correlation between the LRC modes of the CUs belonging to the same quad-tree structure within a CTU; these can be utilized for predicting the LRC mode of the children CUs. In particular, the blue-color CU in Fig. 3.4 is correlated with the tan-color CUs, thus its LRC mode can be predicted based on the LRC modes of tan-color CUs [107].

Here, our objective is to propose a probabilistic model that uses the above-mentioned correlation to predict probabilities of inter/intra modes of the to-be-encoded CU in current DV_t. To this end, we approximate a function from parent/ancestor CU to child CU, or equivalently, a posterior probability of the mode of current CU given the coding information (such as the mode

information) of its parent/ancestor CUs. The estimation of this posterior probability (or equivalently this function approximation problem) can be modeled as a supervised learning problem. As a first step for this function approximation, we build a probabilistic model that utilizes the coding information of the parents/ancestors CUs in the previous quad-tree depth layers in current DV_t, to predict the probability of each mode of the to-be-coded CU. We have also observed that there is correlation between mode of the to-be-coded CU and having AZB in the parent CU. For instance, the chance of having the Merge mode in child CUs is more than 90% on average for our training video sequences, when AZB is detected in the parent CUs. Note that AZB blocks are the residual blocks which all of their elements become equal to zero after taking transform and quantization. In the case that AZB is detected for a CU, AZB=1; Otherwise, AZB=0. Therefore, for mode prediction in the quad-tree parenthood model, we use the mode and AZB information of the parent/ancestor CUs. Hence, a probability is assigned to each mode and AZB of the current CU at the n^{th} dependent texture view (n^{th} DV_t) (in depth *d*) given the mode of the parent/ancestor CU in the previous depth layer (depth *p*) and the same DV_t (i.e., P(mode of current CU at the n^{th} DV_t in depth *d* | mode and AZB of the parent/ancestor CU in depth *p* and the same DV_t)). To define this posterior probability, different numbers are assigned to different 3D-HEVC modes (including the Merge/Skip, inter and intra modes). We call these numbers the mode classes (labels), hereafter.

Assume Y_d is the random variable corresponding to the LRC mode of the current CU in depth *d*, Y_p is a random vector corresponding to the LRC modes of its parent/ancestor CU in the quad-tree depth *p* (where $p<d$), and AZB_p is a random vector corresponding to observing AZB in the parent/ancestor CU in the quad-tree depth *p*. Therefore, the posterior probability can be written as $P_{ndp}(Y_d|Y_p, AZB_p)$ at the n^{th} DV_t. The probability of observing one mode in the current block,

given the probabilities of the modes and their AZB already found in the previous depths, can be computed using Maximum Likelihood Estimation (MLE) [89] as follows:

$$P_{ndp}\left(Y_d = k \middle| Y_p = q,\ AZB_p = z_p\right) = \frac{N_{ndpkqz}}{N_{npqz}}\ ,\ for\ p = 0,1,\dots,d-1 \qquad (4.15)$$

where the element N_{ndpkqz} denotes the number of times that the mode of a CU in the quad-tree depth d and the nth DV$_t$ is k, when the mode of its ancestor/parent in the quad-tree depth p and the same DV$_t$ is equal to q and AZB for that ancestor/parent CU is equal to z_p. N_{npqz} indicates the number of times that the mode of a parent CU in the quad-tree depth p and the nth DV$_t$ is equal to q and its AZB is equal to z_p. Note that in our study the CTUs are of size 64x64 and the maximum coding depth of the quad-tree is equal to 3 (according to the encoding configuration we use in this implementation). To design an appropriate quad-tree parenthood model, we need to compute all the possible conditional probabilities between the mode information of child CUs at different depths given the coding information of parent/ancestor CUs for each DV$_t$. Therefore, six conditional probabilities, P_{n32}, P_{n31}, P_{n30}, P_{n21}, P_{n20}, and P_{n10} are calculated for the nth DV$_t$.

A major drawback of MLE is that when it is used for estimating probabilities, there are some situations in which P_{ndp} becomes equal to zero. This means that we may not encounter some states (modes) during training. Therefore, in this case the classifier will not yield the best possible solution. This is an example of over-fitting [89]. We address this problem by using Maximum a Posteriori (MAP) estimation [91]. MAP estimation is a regularization of MLE, which resolves the above-mentioned over-fitting problem by incorporating a prior distribution over the parameter that we want to approximate. Since our objective is to predict the mode of the current CU, a prior distribution is assigned using the prior knowledge we have about those modes.

One of the main steps in utilizing MAP estimation, is assigning initial probabilities generated from the training process. Since the number of inter-intra prediction modes in 3D-HEVC is larger than two, the distribution of the P_{ndp} here is a categorical distribution. Categorical distribution is a distribution which describes the probabilities of a random event that has M (in this study M is equal to number of modes) outcomes and explains the probabilities separately. That is, the categorical (multinomial) distribution describes the possibility of each of the possible modes. As a result, for the quad-tree parenthood approach, the solution for the MAP estimate is:

$$P_{ndp}\left(Y_d = k \middle| Y_p = q \;\; AZB_p = z_p\right) = \frac{N_{ndpkqz} + \alpha_{ndpkqz}}{N_{npqz} + \sum_{k=1}^{K} \sum_{z=0}^{z=1} \alpha_{ndpkqz}}, \qquad \text{for } p=0,1,\ldots,d\text{-}1 \tag{4.16}$$

where α_{ndpkqz} determines the strength of the prior assumptions relative to observed data, and K is equal to the number of distinct values which Y_p can take. Note that α_{ndpkqz} is the hyper parameter for the quad-tree parenthood model. The hyper parameters are computed during the training process as explained later.

In the quad-tree parenthood model complexity reduction approach, the information of the mode with minimum RD-cost for each CU size and its AZB, during CTU partitioning, are kept in the memory until the coding of the frame is over. This helps us to utilize the coding information of parent/ancestor CUs in previous quad-tree depths when building the quad-tree parenthood probabilistic model for dependent texture views.

In summary, we build a model for the probabilities of all the available modes of the current CU based on the coding information (mode and AZB) relations that exist between the modes of CUs in different quad-tree depths of each CTU. By using this model, it is possible to predict the LRC mode of the to-be-coded block of current DV_t. In the following Subsection we introduce a fast mode assignment for the complexity reduction based on the quad-tree parenthood model.

4.3.1.2 FMA Based on Quad-tree Parenthood Model for Dependent Texture Views

In this Section, we propose a FMA method whose main goal is to gradually update the quad-tree parenthood model and to utilize it to predict the mode with the lowest RD-cost for the CUs in DV_t. The mode-decision making process inside the encoder is content dependent. That is, once the content changes the modes are expected to change too. In addition, modes also depend on the quantization parameters and the chosen encoding configuration. The latter determines the available modes and other related parameters (e.g., search range, GOP, etc.). For instance, in some 3D-HEVC configurations, using skip mode is disabled. In other words, the model and its efficiency will be highly dependent on the configuration used during training. To address this issue, we decided to implement our FMA using an online learning approach. In online learning, the main goal is coming up with an initial prediction model that is updated over time. Similar to other machine learning approaches, the main target is predicting labels (modes with the lowest RD-cost in this case). Hence, in this study we build the initial quad-tree parenthood model using a training video database (see the result Section for more details). During the training process (model initialization), the initial model (hyper parameters) for conditional probabilities (see equation 2) is found. Thus, before starting the training process, the hyper parameter is set to zero. This assumption makes MAP estimation the same as MLE. To build the initial model, the training video dataset is encoded using the original 3D-HEVC encoder and the coding information (modes and AZBs) of all of the blocks (of different sizes) within the same quad-tree structure of the CTUs of the BV_t and dependent texture views are stored in memory. The 3D-HEVC encoder checks all the available inter/intra prediction modes, to find the mode that has the lowest RD-cost for each CU in BV and DVs. Then, if the quad-tree depth of to-be-encoded CU

in current DV_t is larger than zero, the mode information of its ancestors is utilized to build the initial model (see equation (4.15)).

For each new video sequence, the initial probabilities of the model are fine-tuned during the encoding process, resulting in an updated model that is efficiently used for the rest of the video sequence. More precisely, the FMA based on quad-tree parenthood model (see Fig. 4.5) is utilized to decrease the complexity of 3D-HEVC. In this FMA, if the quad-tree coding depth of the to-be-coded CU is larger than zero, based on the mode and AZB of the parent/ancestor CUs in the previous quad-tree depth/depths and the quad-tree parenthood model (using equation (4.16)) four most probable modes are found. These modes are suggested as the modes that are more likely to be the LRC mode. The probability of these modes should be very high (more than 0.95). In Section 4.1.2, an RD-cost threshold (Thr) is suggested for each of the to-be-encoded DV_t CUs by taking into consideration the disparity between the base view and the current DV_t. This threshold is computed using the RD-cost values of already encoded neighbouring blocks in the current DV_t and their corresponding BV_t blocks. In order to increase the chance of selecting the right mode (LRC mode), we use the RD-cost threshold proposed in Section 4.1.2. More precisely, if the probability of at least one of the modes or together (given the LRC mode and AZB of parent CU) is larger than 0.95, and at least the RD-cost value of one of the suggested modes is not greater than the threshold, the mode candidate which has the lowest RD-cost among the mode candidates is suggested as the LRC mode for current DV_t CU. However, if the mode candidates cannot fulfill the probability condition or the RD-cost condition, the FMA cannot find the LRC mode. Note, the RD-cost threshold is not defined for the cases that less than two neighbors are available (see Section 4.1.2). Therefore, in this case the quad-tree parenthood model FMA fails to find the LRC mode. In addition, when the quad-tree depth is equal to zero,

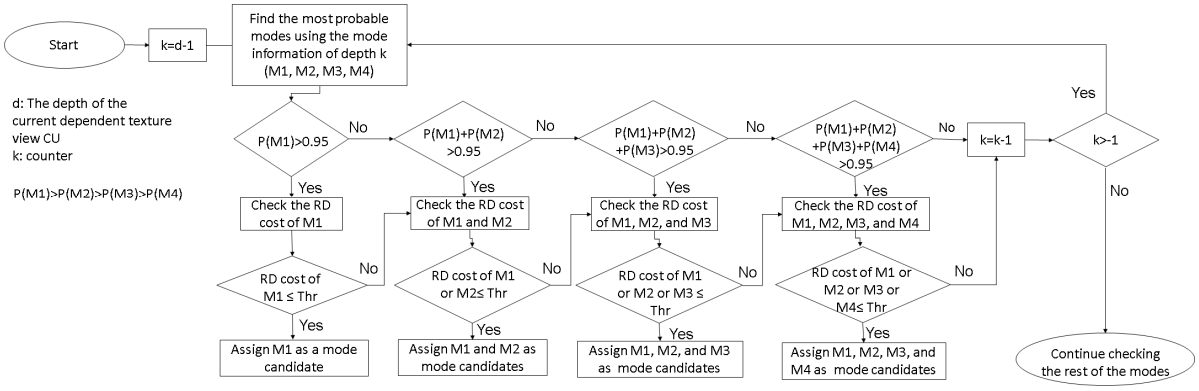

Figure 4.5 The block diagram of FMA based on quad-tree parenthood model.

this FMA cannot be utilized. In all the cases that the FMA fails to find the LRC mode, the encoder checks the unchecked modes to find the LRC mode.

As can be seen in Fig. 4.5, this parenthood model FMA may check one to four modes based on the four probabilistic rules (probability condition) and four RD-cost rules (threshold). Therefore, this approach decreases the computational complexity of encoding DV_t significantly.

4.3.2 Mode Search Reduction Using Neighborhood Model for Dependent Texture Views

4.3.2.1 Neighborhood Model (Bayesian Approach) for Dependent Texture Views

In Section 4.2, a mode prediction method was proposed which used the LRC modes of the four neighbouring blocks in the current DV_t and the LRC mode of the corresponding block in the BV_t to predict the LRC mode of to-be-encoded block in the current DV_t. We called this five CUs as the predictor CUs (see Section 4.2). In that neighborhood model implementation, the probability of each mode of the current CU was computed during the training process and used

later. In order to improve the mode prediction accuracy of the method proposed in Section 4.2, more coding information of already encoded CUs are used in our new method. In this study, we add the co-located block in the previous frame of the same DV_t to the predictor CUs. We call this predictor CU as the temporal predictor CU. In this Section, the temporal predictor CU is our first predictor CU in the predictor CUs list. The second predictor CU for the current DV_t CU is its corresponding CU in the BV_t. We call this the BV_t predictor CU, hereafter. This new block is added to take the advantage of temporal correlation which exists in each view frame. Moreover, in the prediction, the new model also considers if the predictor CUs are AZB blocks or not. Our studies show that motion vector information of the predictor CUs can also be used for mode prediction. We have observed that when the corresponding CU in the BV_t belongs to the region with homogeneous motion, the chance of having the merge mode as the LRC mode of current CU in the current DV_t is more than 90% for the GhostTownFly video sequence. Our statistical studies also show the chance of having the merge mode as the LRC mode for a DV_t CU is more than 85% on average for our training video sequences, when the temporal predictor CU belongs to the region with homogenous motion. In order to extract more complex relations between the DV_t modes and the motion information of the BV_t predictor CU and the temporal predictor CU, we add the motion homogeneity of those two predictor CUs into features that are used for mode prediction. In [40, 41], an approach is suggested to classify each CTU into three Motion Homogeneity Categories (MHCs): 1) with homogenous motion, 2) with moderate motion, and 3) with complex motion. In this study, the motion homogeneity classification approach proposed in Section 3.1.2 (of Chapter 3) is used to classify motion homogeneity of the predictor CTUs. Then, the same MHC is assigned for all the CUs with in each CTU. Note, in this study, the MHCs of the neighbouring blocks are not used for the mode prediction. Because only in the case

that the frame is encoded, the motion vectors of all the CUs become available and we can classify each CU in terms of motion homogeneity. In order to make model that uses the LRC mode and AZB of the predictor CUs, and MHC of the first (temporal) and second (BV_t) predictor CUs for mode prediction, a probability is assigned to each mode of the current CU at the n^{th} DV_t given the modes, AZBs, and MHCs of its predictor CUs (i.e. P(mode of current CU at the n^{th} DV_t| modes and AZBs of the predictor CUs, and MHCs of the BV_t and temporal predictor CUs)). Assume *Y* is the random variable corresponding to the mode of the current CU, *X* is a random vector corresponding to the modes of its predictor CUs, *AZB* is a random variable corresponding to observing *AZB* in the predictor CUs, and *MHC* is a random vector corresponding to the *MHC*s of the first and second predictor CUs. The posterior probability $P_n(Y|X, AZB, MHC)$ (which determined the probability of the mode of the to be encoded CU at the n^{th} DV_t given the mode, AZB, and motion of the predictor CUs) can be calculated using the Bayes rule:

$$P_n(Y|X, AZB, MHC) = \frac{P_n(X, AZB, MHC|Y)P_n(Y)}{\sum_Y P_n(Y)P_n(X, AZB, MHC|Y)} \quad (4.17)$$

To train a classifier each of $P_n(X, AZB, MHC|Y)$ and $P_n(Y)$ should be estimated. $P_n(Y)$ is our prior for the to-be encoded CU at the n^{th} DV_t. Suppose that there are *M* different values for different 3D-HEVC modes which would result in M different values for the random variable *Y*. Also, *X* is a vector with *L* components (number of predictor CUs) each taking *M* possible discrete values which will result in $M^L - 1$ different values for the random vector *X*. *AZB* is a vector with *L* components each taking *Z=2* discrete values. *MHC* is a vector with two (temporal and BV_t predictor CUs) components each taking *H=3* discrete values. Therefore, random vectors AZB and *MHC* have 2^L-1 and 8 different values. In this study, *L* is equal to the number of predictor CUs (*L=6*). The learning algorithm needs to estimate *M-1* different parameters to

estimate $P_n(Y)$, because the probability should sum to one. However, estimating $P_n(X, AZB, MHC|Y)$ requires learning of an exponential number of parameters i.e. $8M(M^L-1)(2^L-1)$, which is an intractable problem. Thereby, the key to use Bayes rule is to specify a suitable model for $P_n(X, AZB, MHC|Y)$.

In our study to solve the above-mentioned intractability problem, Naive Bayes classifier [89], [91] has been used. Naive Bayes classifier dramatically reduces the complexity of estimating $P_n(X,AZB,MHC|Y)$ by making a conditional independence assumption between the information of the predictor CUs. This learning algorithm assumes that different members of the X vector and their AZB and MHC are independent given Y. Taking into account the conditional independence assumption we have:

$$P_n(X, AZB, MHC|Y) = P_n((X_1, AZB_1, MHC_1), (X_2, AZB_2, MHC_2),$$

$$(X_3, AZB_3) \dots, (X_L, AZB_L)|Y)$$

$$= \prod_{l=1}^{2} P_n(X_l, AZB_l, MHC_l|Y) \prod_{l=3}^{L} P_n(X_l, AZB_l|Y) \quad (4.18)$$

Therefore,

$$P_n(Y|X, AZB, MHC) = \frac{\prod_{l=1}^{2} P_n(X_l,AZB_l,MHC_l|Y) \prod_{l=3}^{L} P_n(X_l,AZB_l|Y)\ P_n(Y)}{\sum_Y P_n(Y) \prod_{l=1}^{L} P_n(X_l,AZB_l,MHC_l|Y) \prod_{l=3}^{L} P_n(X_l,AZB_l|Y)} \quad (4.19)$$

This simplifying assumption makes the representation of $P_n(X,AZB,MHC|Y)$ simpler and reduces the number of parameters from an exponential term to just M(M*12)+M(2M(L-2)).

To find the $P_n(Y|X,AZB,MHC)$, we need to have $P_n(X,AZB,MHC|Y)$ and $P_n(Y)$. Because of the major drawback of MLE, we have decided to use MAP estimation [91], [107]. The solution to the MAP estimate for $P(Y)$ is:

$$P_n(Y = y_k) = \frac{N_{nk}+\alpha_{nk}}{N_{nT}+\sum_{k=1}^{M} \alpha_{\mathrm{nk}}} \quad (4.20)$$

where the element N_{nk} is the number of observed instances of class y_k. That is, N_{nk} indicates the number of times the modes of the current CU at the n^{th} DV_t is equal to y_k. N_{nT} shows total number

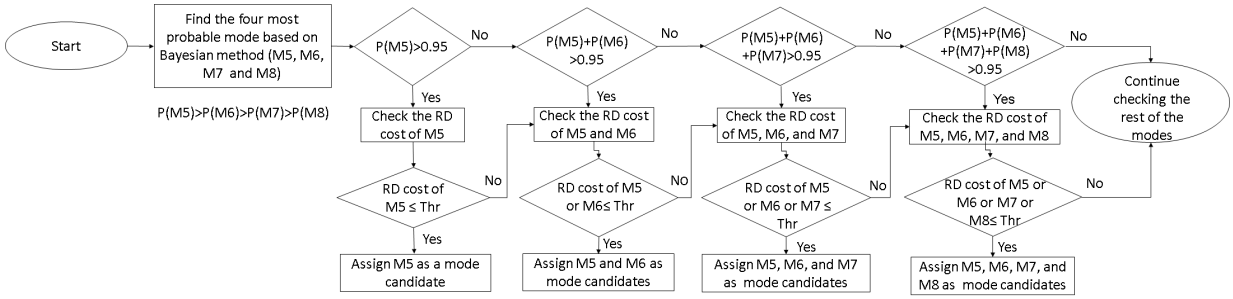

Figure 4.6 The block diagram of FMA based on neighborhood model.

of CUs which were encoded in the nth DV$_t$. α_{nk} determines the strength of the prior assumptions relative to the observed data and M is equal to the number of different values which Y can take. On the other hand for the temporal and BV$_t$ predictor CUs the estimate for $P_n(X, AZB, MHC|Y)$ is as follows:

$$P_n(X_l = x_{lm}, AZB_l = z_{lz}, MHC_l = h_{lh}|Y = y_k) = \frac{N_{nlmzhk}+\alpha_{nlmzh}}{N_{nk}+\sum_{m=1}^{M}\sum_{z=0}^{z=1}\sum_{h=1}^{h=3}\alpha_{\mathrm{nlmzh}}}, \text{ for l=1,2} \quad (4.21)$$

where the element N_{nlmzhk} denotes the number of times $X_l=x_{lm}$, $AZB_l=z_{lh}$, and $MHC_l = h_{lh}$ have been observed in the instances of class y_k. That is, N_{nlmzhk} indicates the number of times the mode of the l^{th} predictor is equal to $x_{lm,}$ and its $AZB=Z_l$, and $MHC_l = h_{lh}$, while the mode of the current CU at the nth DV$_t$ is equal to y_k. Here, the term N_{nk} is the number of time the mode y_k has been observed in the nth DV$_t$. α_{nlmzh} determines the strength of the prior assumptions relative to the observed data and M is equal to the number of distinct values which X_l can take.

For the neighbouring predictor CUs the estimate for $P_n(X, AZB|Y)$ is as follows:

$$P_n(X_l = x_{lm}, AZB_l = z_{lz}|Y = y_k) = \frac{N_{nlmzk}+\alpha_{nlmz}}{N_{nk}+\sum_{m=1}^{M}\sum_{z=0}^{z=1}\alpha_{\mathrm{nlmz}}} \quad \text{, for l=3,…,L} \quad (4.22)$$

where the element N_{nlmzk} denotes the number of times $X_l=x_{lm}$ and $AZB_l=z_{lh}$ $MHC_l = h_{lh}$ have been observed in the instances of class y_k. That is, N_{nlmzk} indicates the number of times the mode of the l^{th} predictor is equal to $x_{lm,}$ and its $AZB=Z_l$, while the mode of the current CU at the nth DV$_t$ is equal to y_k. Here, α_{nlmz} determines the strength of the prior assumptions relative to the observed

data.

Unlike the method proposed in Section 4.2, in this work we implement the test process following on online-learning. More precisely, the neighborhood model is fine-tuned during the course of encoding. We use four probabilistic rules and four RD-cost rules to propose an FMA based on the neighborhood model. By using these rules, this FMA is able to check a smaller number of modes to find the mode with the lowest RD-cost for the CUs in each DV_t compared to the method proposed in Section 4.2. In the following Subsection, we present the FMA based on the neighborhood model.

4.3.2.2 FMA Based on Neighborhood Model for Dependent Texture Views

The objective here is to implement a fast mode assignment based on the neighborhood model, so that the encoder does not need to go through all the modes, thus significantly reducing the computational complexity of the mode-search process of the 3D-HEVC. Here, we propose a FMA that fine-tunes the neighborhood model during the encoding process while utilizing that model to predict the LRC mode of to-be-coded CUs in each DV_t. In order to predict the LRC mode of current CU at the n^{th} DV_t, it is suggested to compute P(mode of current CU at the n^{th} DV_t| modes, AZBs of predictor CUs, and motion homogeneity of the BV_t and temporal predictor CUs). The estimation of this posterior probability is modeled as a supervised learning problem.

Our supervised learning problem consists of two stages; the first stage is the training process and the second stage is the test-tune process. Same as the parenthood model, during the training process (initializing model) of the neighborhood model, our training video sequences are encoded using the unmodified 3D-HEVC. The mode information will be used to make the initial model (compute hyper parameters α_{nk} (Equ. 4.20), α_{nlmzh} (Equ. 4.21), and α_{nlmz} (Equ. 4.22)) for

the CUs at the n^{th} DV_t [107]. There are some circumstances in which the information about all neighbouring predictor CUs is not available. In such cases the mode of the neighbouring CUs and the co-located block in the previous frame (temporal CU) along with the corresponding CU in the BV_t is used to build the model. If all four neighbors of a CU in the current DV_t are not available, the model just uses the BV_t and temporal CU information as a predictor (if available). Otherwise, the neighborhood probabilistic model is not updated.

During the test-tuning process, the unmodified 3D-HEVC encoder is used to encode the BV of the training video sequences. Then, the neighborhood model is employed to predict the probabilities of all the available modes for the to-be-coded CUs in DV_t. Fig. 4.6 illustrates the block diagram of the proposed neighborhood model based FMA. Here, if at least one of the predictor CUs is available, the four most probable modes are found using the optimal Bayesian decision approach [107]. In order to suggest LRC modes for the to-be-encoded CU in the current DV_t, two conditions are considered. First, the probability of the suggested modes needs to be larger than 0.95, thus resulting in insignificant bite-rate increase. As discussed before, this FMA uses an online learning approach to gradually build the probabilistic model for the CUs in current DV_t. Therefore, the chance of choosing wrong modes for CUs, especially for the CUs of the first frames, is not low. As a result, same as the parenthood model, the RD-cost threshold proposed in Section 4.1.2 is utilized to form the second condition. Note that by using the probability condition and the RD-cost condition it is possible to check less number of modes compared to the method proposed in Section 4.2 to find the LRC mode. First, the RD-cost threshold is computed using the available neighbouring CUs (see Section 4.1.2). Then, the FMA checks two conditions: If at least one of suggested modes or together have the probability which is larger than 0.95 and the RD-cost value of at least one of them is not greater than the RD-cost threshold

(proposed in Section 4.1.2), the fast mode assignment based on the neighborhood model will be applied and the rest of the modes won't be checked. If the mode candidates do not fulfill the probability condition or the RD-cost condition, this FMA is not able to find the mode. Note that if fewer than two neighbors are available, the threshold value is not available. In the case that the FMA fails to find the mode, the unchecked modes will be checked by the encoder to find the LRC mode. Finally, the neighborhood model (probabilities) will be updated using the LRC mode of current CU in the current DV_t. As can be seen in Fig. 4, the neighborhood model FMA may check one to four modes based on the four probabilistic rules and the four RD-cost rules. Therefore, the complexity reduction based on the neighborhood model leads to significant computational complexity reduction for encoding CUs in dependent texture views.

4.3.3 Online-learning Based Hybrid Complexity Reduction Scheme for Dependent Texture Views

In the previous Subsections, to decrease the mode search process in dependent texture views two FMAs were proposed. To further decrease the complexity of 3D-HEVC, one may consider reducing the inter prediction search window and implement early termination mode search approaches. In our final design of a hybrid complexity reduction scheme, we utilize a suitable combination of the FMA approaches described above, our early termination method proposed in Section 4.1.2, and our adaptive search range adjustment method also proposed in Section 4.1.1. This combination leads to an effective scheme with the best possible complexity reduction performance. Fig. 4.7 shows the block diagram of our hybrid complexity reduction scheme. First, the probabilistic quad-tree parenthood model and probabilistic neighborhood model are initialized using the training video sequences. More precisely, our training video sequences are

encoded using unmodified 3D-HEVC encoder and the mode information of the encoded CUs is used to build the model. For any new video sequence the BV is encoded using the unmodified 3D-HEVC encoder. The mode and motion information of the CUs in BV_t are stored in memory. The BV_t motion information is utilized later for predicting the appropriate search range for each CTU in current DV_t using the adaptive search range adjustment method proposed in Section 4.1.1. Afterwards, the encoder starts encoding each DV. Unlike the training process, the encoder does not check all of the inter-intra prediction modes for the current DV_t CU. Instead, if the coding depth of the current CU is equal to zero, the neighborhood model FMA is utilized to predict the mode. If this FMA approach fails to predict the mode, the encoder starts checking the unchecked modes and once the RD-cost value of one of the checked modes is lower than a predetermined threshold (see Section 4.1.2), the encoder applies the early termination method and the rest of the modes are checked. For cases in which the quad-tree coding depth is greater than zero, first the hybrid complexity reduction scheme uses the parenthood model FMA. If this FMA fails to find the LRC mode, the neighborhood model FMA is utilized to predict the best

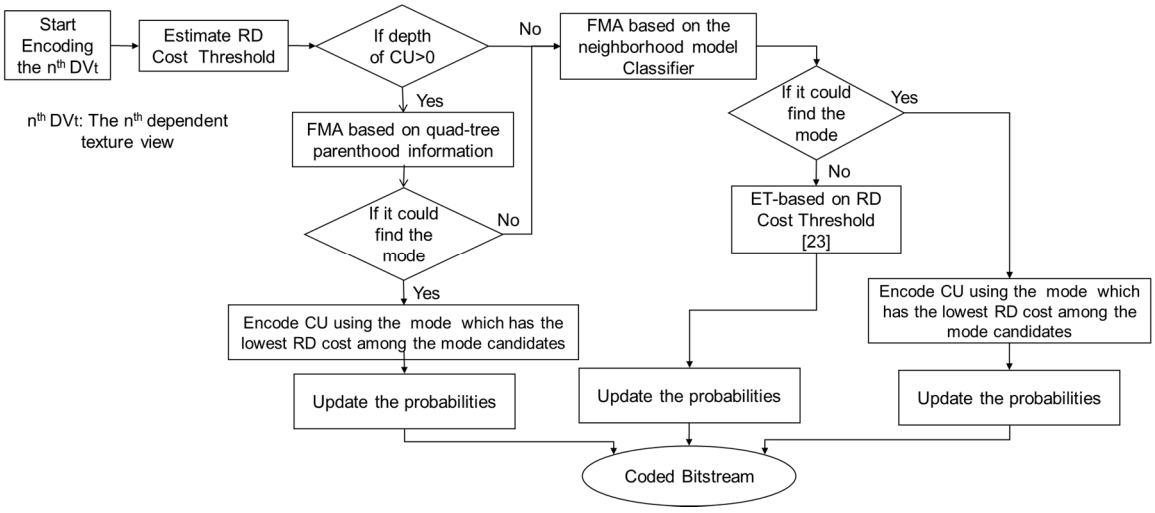

Figure 4.7 Block diagram of our proposed Hybrid complexity reduction scheme for the dependent texture views of the 3D-HEVC encoder.

mode. If this FMA is not able to find the best mode, the hybrid complexity reduction scheme uses the early termination method proposed in Section 4.1.2. If the RD-cost of one of the checked modes is not greater than the threshold, this mode is selected as the LRC mode by the ET method. Finally, the proposed hybrid method will update the two models using the LRC mode of the current block.

In online-learning based hybrid complexity reduction method similar to the FMAs, online learning approach is utilized. This is because it is expected to see scene and content variations over time during the encoding process of a video. Therefore, we need to update the model during the encoding process. In other words, the probabilistic models will be fine-tuned gradually.

4.4 Experimental Results and Discussions

In our experiment, we use three training video sequences, the Champagne_tower [108], Pantomime [108], and Love-bird1 [109]. The resolution, frame rate, and the views of the training video sequences are described in Table 4.2. In addition, eight test videos from the data set proposed by the common test conditions (CTCs) [110]–[114] were used for validating our method (see Table 4.3) and all simulations were performed under CTCs. Note that the training dataset used for finding the hyper parameters was not included in our validating test videos. Our method was implemented in the 3D-HEVC software (HTM-12.2). In this software the fast encoder decision for texture coding [80] was enabled. For simplicity reasons we test the case

Table 4.2 Training video dataset specifications.

Name	Resolution, Frame Rate (fps)	Base & Dependent Views
Champagne_tower	1280x960,30	V_{37} , V_{39}
Pantomime	1280x960,30	V_{40} , V_{42}
Love-bird1	1024x768, 30	V_{6} , V_{8}

with only two views (BV+DV). More precisely, the ‘’baseCfg_2view+depth” configuration of 3D-HEVC encoder [115] is used (hierarchal B pictures and GOP length 8). The QPs used for the views and the depth (QpV,QpD) are as follows: (25, 34), (30,39), (35,42) and (40, 45). The performance of our proposed hybrid scheme is compared with the complexity reduction methods proposed in Section 4.1 and Section 4.2. Considering that our hybrid scheme consists of several approaches, in our experiment multiple testing scenarios are taken into account to evaluate the effectiveness of different approaches. In our experiment we evaluate the performance of 1) our parenthood model FMA (presented in Section 4.3.1), 2) our neighborhood model FMA (presented in Section 4.3.2), 3) Hybrid complexity reduction scheme without online learning (to investigate the effect of disabling the online learning approach), 4) Online-learning hybrid complexity reduction with Laplace smoothing (to investigate the effect of using different hyper parameters), 5) online-learning complexity reduction method without RD cost threshold (when RD cost threshold conditions are not checked in Fig. 4.5 and Fig. 4.6), 6) online-learning complexity reduction method without the probability conditions (when probability conditions in Fig. 4.5 and Fig. 4.6 are not checked), and 7) complete version of the online-learning based hybrid complexity reduction scheme (presented in Section 4.3.3 and Fig. 5). Note, in the third testing scenario (hybrid scheme without online-learning), the probabilistic models are not updated during the encoding process of the new video sequence (the fine-tune process is disabled). More precisely, the two probabilistic models are built using the training video sequences. Then, for a new video sequence these models are used for mode prediction. In the fourth testing scenario we investigate the sensitivity of our online learning scheme to the training data and the prior knowledge that is used to set the hyper parameters. To this end, instead of using the information of the training video sequences to set the hyper parameters, we assign

Table 4.3 Test video dataset specifications.

Name	Resolution, Frame Rate (fps)	Base & Dependent Views
Kendo	1024x768, 30fps	V_1 , V_3
Newspaper1	1024x768, 30fps	V_2 , V_4
Poznan_Hall2	1920x1088, 25fps	V_7 , V_6
GhostTownFly	1920x1088, 25fps	V_9 , V_5
Balloons	1024x768, 30fps	V_1 , V_3
Poznan_street	1920x1088, 25fps	V_5 , V_4
Shark	1920x1088, 30fps	V_1 , V_5
Undo_Dancer	1920x1088, 25fps	V_1 , V_5

Dirichlet prior distribution over our hyper parameters, with the equal valued parameters [89]. Here we use Laplace smoothing in which all the hyper parameters are equal to one [89]. By using the Laplace smoothing, in the beginning of the test process of the online learning scheme all the inter/intra modes have the same probabilities and the probabilistic models are fine-tuned gradually in the course of encoding. In the fifth and the sixth testing scenarios the effect of the probability conditions as well as the RD cost threshold conditions on our online-learning hybrid scheme are investigated. In the online-learning complexity reduction without the probability conditions (sixth testing scenario), the FMAs presented in Section 4.3.1.2 and Section 4.3.2.2 suggest the four most probable modes and only check the RD cost threshold condition. If at least one of the candidate modes fulfills the RD cost threshold condition, the modified encoder won't check the rest of the modes.

The results of our experiments are reported in Table 4.4 and Table 4.5. Table 4.4 shows the impact of different methods on the Bjøntegaard-Delta bit-rate (BD-BR) values [99] as suggested in CTCs [99] for video 1 (BD-BR of dependent view), video PSNR/video bitrate (BD-BR of video0 + video1), video PSNR/total bitrate (BD-BR of video0+video1+depth maps) and synthesized PSNR/total bitrate (BD-BR of synthesized views). Table V reports the coding execution time reduction percentage and accuracy of the mode prediction for the DV_t compared to unmodified 3D-HEVC for the video streams.

As it is observed from Table 4.5, the combination of the adaptive search range adjustment and ET method (Section 4.1) reduces the DV_t coding execution time on average by 29.27% compared to unmodified 3D-HEVC encoder, at the cost of 1.98% average bitrate-increase (see Table 4.4) for video PSNR/total bitrate BD-BR and 1.91% average bitrate increase for the synthesized PSNR/total bitrate BD-BR (see Table 4.4). For this method, the average mode prediction accuracy for the DV_t is about 88.10% (see Table 4.5). As illustrated in Table 4.4 and

Table 4.4 The impact of all the methods on bitrate (of views and synthesized views) for the test video sequences.

Methods Tested in Our Study	BD-BR	Test Video Sequences								
		Kendo	Newspaper1	Poznan_Hall2	GhostTownFly	Balloons	Poznan_street	Shark	Undo-Dancer	**Average BD-BR**
Content adaptive complexity reduction scheme (Adaptive search Range adjustment + ET)	Video 1	4.31%	4.54%	6.35%	6.71%	8.30%	8.13%	4.22%	3.81%	**5.80%**
	Video PSNR/Video Bitrate	1.54%	1.67%,	2.32%	2.51%	2.72%	2.31%	1.35%	1.11%	**1.98%**
	Video PSNR/Total Bitrate	1.53%,	1.65%	2.32%	2.49%	2.71%	2.30%	1.31%	1.10%	**1.98%**
	Synth PSNR/Total Bitrate	1.49%	1.63%	2.35%	2.42%	2.69%	2.31%	1.29%	1.08%	**1.91%**
Low complexity mode decision approach for 3D-HEVC	Video 1	4.42%	2.18%	2.99%	2.25%	3.20%	3.51%	3.95%	1.85%	**3.04%**
	Video PSNR/Video Bitrate	1.58%	0.71%	1.01%	0.83%	0.85%	0.88%	1.22%	0.75%	**0.98%**
	Video PSNR/Total Bitrate	1.58%	0.68%	0.99%	0.82%	0.84%	0.87%	1.21%	0.74%	**0.97%**
	Synth PSNR/Total Bitrate	1.60%	0.71%	0.98%	0.79%	0.82%	0.86%	1.19%	0.67%	**0.95%**
FMA based on quad-tree parenthood model	Video 1	1.12%	1.24%	0.96%	0.44%	0.81%	0.71%	0.69%	0.45%	**0.80%**
	Video PSNR/Video Bitrate	0.29%	0.23%	0.28%	0.12%	0.21%	0.19%	0.21%	0.10%	**0.20%**
	Video PSNR/Total Bitrate	0.28%	0.22%	0.27%	0.11%	0.19%	0.17%	0.19%	0.09%	**0.19%**
	Synth PSNR/Total Bitrate	0.23%	0.16%	0.21%	0.09%	0.14%	0.11%	0.17%	0.08%	**0.15%**
FMA based on neighbourhood model	Video 1	1.37%	1.53%	0.95%	0.59%	1.19%	0.59%	0.83%	0.64%	**0.96%**
	Video PSNR/Video Bitrate	0.35%	0.32%	0.27%	0.17%	0.32%	0.15%	0.24%	0.18%	**0.25%**
	Video PSNR/Total Bitrate	0.33%	0.31%	0.27%	0.16%	0.29%	0.14%	0.24%	0.17%	**0.24%**
	Synth PSNR/Total Bitrate	0.30%	0.29%	0.22%	0.12%	0.25%	0.12%	0.22%	0.13%	**0.21%**
Hybrid complexity reduction method without online-learning	Video 1	2.98%	2.86%	2.52%	2.45%	3.40%	2.51%	3.32%	2.71%	**2.84%**
	Video PSNR/Video Bitrate	0.93%	0.94%	0.83%	0.88%	0.89%	0.67%	0.84%	1.01%	**0.87%**
	Video PSNR/Total Bitrate	0.92%	0.93%	0.81%	0.86%	0.87%	0.65%	0.83%	0.98%	**0.86%**
	Synth PSNR/Total Bitrate	0.91%	0.94%	0.81%	0.82%	0.84%	0.64%	0.82%	0.94%	**0.84%**
Online-learning based hybrid complexity reduction method with Laplace smoothing	Video 1	1.98%	2.01%	1.30%	0.74%	1.50%	1.09%	1.28%	0.93%	**1.35%**
	Video PSNR/Video Bitrate	0.56%	0.60%	0.43%	0.22%	0.44%	0.36%	0.42%	0.27%	**0.41%**
	Video PSNR/Total Bitrate	0.53%	0.58%	0.40%	0.21%	0.41%	0.35%	0.40%	0.25%	**0.39%**
	Synth PSNR/Total Bitrate	0.50%	0.55%	0.37%	0.21%	0.38%	0.33%	0.38%	0.24%	**0.37%**
Online-learning based hybrid complexity reduction method without RD cost threshold	Video 1	2.23%	1.82%	1.28%	1.02%	1.84%	1.60%	1.84%	1.35%	**1.62%**
	Video PSNR/Video Bitrate	0.65%	0.50%	0.41%	0.32%	0.61%	0.58%	0.62%	0.50%	**0.52%**
	Video PSNR/Total Bitrate	0.64%	0.49%	0.39%	0.31%	0.59%	0.56%	0.59%	0.48%	**0.51%**
	Synth PSNR/Total Bitrate	0.63%	0.49%	0.38%	0.30%	0.58%	0.54%	0.58%	0.46%	**0.50%**
Online-learning hybrid complexity reduction method without the probability conditions	Video 1	4.41%	4.72%	6.15%	6.01%	8.78%	5.23%	4.61%	4.54%	**5.56%**
	Video PSNR/Video Bitrate	1.58%	1.77%	2.19%	2.02%	3.01%	1.79%	1.59%	1.39%	**1.92%**
	Video PSNR/Total Bitrate	1.55%	1.76%	2.14%	1.99%	2.99%	1.77%	1.57%	1.36%	**1.89%**
	Synth PSNR/Total Bitrate	1.54%	1.77%	2.09%	1.99%	2.95%	1.74%	1.55%	1.32%	**1.87%**
Complete version of the online-learning based hybrid complexity reduction method	Video 1	**1.51%**	**1.62%**	**1.04%**	**0.76%**	**1.32%**	**0.99%**	**1.04%**	**0.95%**	**1.15%**
	Video PSNR/Video Bitrate	**0.39%**	**0.38%**	**0.33%**	**0.23%**	**0.39%**	**0.44%**	**0.36%**	**0.31%**	**0.35%**
	Video PSNR/Total Bitrate	**0.38%**	**0.38%**	**0.31%**	**0.22%**	**0.35%**	**0.43%**	**0.35%**	**0.29%**	**0.34%**
	Synth PSNR/Total Bitrate	**0.36%**	**0.36%**	**0.26%**	**0.21%**	**0.31%**	**0.41%**	**0.31%**	**0.22%**	**0.31%**

Table 4.5, if the parenthood model FMA is used, the coding DV_t execution time is reduced on average by 44.66% compared to the unmodified 3D-HEVC encoder at the cost of 0.19% average bitrate increase for the video PSNR/total bitrate and 0.15% average bitrate increase for the synthesized PSNR/total bitrate BD-BR. This method achieves the DV_t mode prediction accuracy of 96.88% on average. When the neighborhood model FMA is used, DV_t execution time reduction percentage of 50.43% to 55.63% is achieved compared to the 3D-HEVC encoder at the cost of 0.24% average bitrate increase for the video PSNR/total bitrate BD-BR and 0.21% average bitrate increase for the synthesized PSNR/total bitrate BD-BR. For this method the DV_t mode prediction accuracy of 96.75% is achieved on average. As it is observed in Table 4.4 and Table 4.5 the performance of the method proposed in Section 4.2 (Low complexity mode decision approach for 3D-HEVC) is better than the combination of adaptive search range adjustment and the ET method (Section 4.1). However, its performance is lower than that of the quad-tree parenthood model FMA and the neighborhood model FMA.

We can also observe from Table 4.4 and Table 4.5 that the complete version of the online-leaning based hybrid scheme outperforms all other methods, achieving DV_t execution time reduction of 67.7% at a mere 0.34% bitrate increase on average for the video PSNR/total bitrate BD-BR and 0.31% bitrate increase for the synthesized PSNR/total bitrate. The average mode prediction accuracy of this method for DV_t is 96.57%. The DV_t time reduction performance of our online-learning based hybrid scheme is on average 25.74% better than our best previously method proposed in Section 4.2. When the fine-tuning process is disabled (no online-learning is involved), the hybrid complexity reduction method decreases the DV_t execution time by 63.37% on average at the cost of 0.86% bitrate increase for the video PSNR/total bitrate BD-BR and 0.84% for the synthesized PSNR/total bitrate BD-BR (see Table 4.4 and Table 4.5). These

results show the importance of our online-learning approach and the reason for including it in our final hybrid method. When the online-learning based hybrid method uses Laplace smoothing, the average DV_t mode prediction accuracy is reduced only by 0.52% compared to that of the complete version. This result shows that using Laplace smoothing for the hyper parameters, the probability conditions and the RD cost threshold conditions in the online-leaning hybrid scheme

Table 4.5 The impact of all the methods on execution time reduction (TR) and the DV_t mode prediction accuracy for the test video sequences.

Methods Tested in Our Study	Time Reduction (TR) & Accuracy	Test Video Sequences								
		Kendo	Newspaper1	Poznan_Hall2	GhostTownFly	Balloons	Poznan_street	Shark	Undo-Dancer	**Average**
Content adaptive complexity reduction scheme (Adaptive search Range adjustment + ET)	Video 1 (DVt) TR	27.11%	24.05%	29.11%	38.22%	31.33%	29.12%	25.11%	30.11%	**29.27%**
	Videos TR	17.90%	17.87%	18.87%	20.46%	20.44%	18.74%	15.92%	18.95%	**18.64%**
	Videos+depths TR	9.42%	8.67%	10.16%	12.91%	10.89%	10.49%	6.02%	9.13%	**9.71%**
	Accuracy	90.82%	90.63%	87.45%	86.87%	83.85%	84.58%	89.79%	90.80%	**88.10%**
Low complexity mode decision approach for 3D-HEVC	Video 1 (DVt) TR	41.92%	40.11%	43.00%	43.81%	43.24%	43.11%	40.21%	40.28%	**41.96%**
	Videos TR	27.90%	29.81%	27.88%	23.45%	28.21%	27.74%	26.63%	25.24%	**27.11%**
	Videos+depths TR	15.23%	15.14%	15.49%	14.94%	15.41%	16.02%	10.74%	12.49%	**14.43%**
	Accuracy	90.55%	93.11%	92.87%	92.11%	91.69%	91.22%	91.34%	93.05%	**91.99%**
FMA based on quad-tree parenthood model	Video 1 (DVt) TR	45.68%	45.59%	43.18%	47.24%	46.25%	43.40%	42.73%	43.19%	**44.66%**
	Videos TR	29.36%	33.81%	27.99%	25.29%	30.19%	27.94%	27.21%	27.90%	**28.71%**
	Videos+depths TR	16.08%	17.30%	15.55%	16.19%	16.56%	16.14%	11.00%	13.91%	**15.34%**
	Accuracy	96.54%	95.52%	96.71%	97.51%	97.27%	97.03%	97.11%	97.32%	**96.88%**
FMA based on neighbourhood model	Video 1 (DVt) TR	52.25%	51.36%	52.72%	55.63%	55.15%	53.30%	51.02%	50.43%	**52.73%**
	Videos TR	33.52%	38.12%	34.18%	29.79%	35.98%	34.30%	32.11%	32.32%	**33.79%**
	Videos+depths TR	18.50%	19.64%	20.21%	19.25%	19.93%	20.04%	13.16%	16.27%	**18.37%**
	Accuracy	96.28%	95.47%	96.73%	97.21%	97.05%	97.16%	96.92%	97.14%	**96.75%**
Hybrid complexity reduction method without online-learning	Video 1 (DVt) TR	64.13%	63.35%	63.02%	66.20%	63.51%	65.22%	62.08%	59.45%	**63.37%**
	Videos TR	41.16%	47.59%	40.87%	38.31%	41.45%	42.02%	41.32%	37.93%	**41.33%**
	Videos+depths TR	21.95%	23.76%	22.17%	24.04%	22.12%	23.77%	16.22%	18.27%	**21.54%**
	Accuracy	92.42%	92.68%	93.21%	91.92%	91.52%	92.97%	92.01%	92.04%	**92.35%**
Online-learning based hybrid complexity reduction with Laplace smoothing	Video 1 (DVt) TR	66.77%	64.49%	64.47%	71.06%	66.85%	67.01%	64.26%	62.65%	**65.95%**
	Videos TR	43.35%	48.06%	42.41%	40.73%	44.04%	43.48%	42.48%	40.20%	**43.09%**
	Videos+depths TR	23.22%	24.02%	23.08%	25.69%	23.62%	24.67%	16.73%	19.48%	**22.56%**
	Accuracy	94.93%	94.21%	95.84%	97.15%	96.47%	96.61%	96.21%	96.99%	**96.05%**
Online-learning based hybrid complexity reduction without RD cost threshold	Video 1 (DVt) TR	66.93%	65.23%	65.15%	70.43%	67.59%	67.52%	66.20%	62.75%	**66.48%**
	Videos TR	43.32%	49.12%	42.69%	40.50%	44.31%	44.26%	42.95%	40.18%	**43.42%**
	Videos+depths TR	23.20%	24.59%	23.25%	25.53%	23.78%	25.15%	16.94%	19.47%	**22.74%**
	Accuracy	94.36%	94.56%	95.89%	96.32%	95.61%	95.35%	95.49%	96.20%	**95.47%**
Online-learning based hybrid complexity reduction method without the probability conditions	Video 1 (DVt) TR	50.95%	45.52%	48.58%	42.47%	47.10%	47.48%	39.30%	39.74%	**45.14%**
	Videos TR	32.73%	33.91%	31.51%	24.18%	30.65%	30.62%	25.66%	25.43%	**29.34%**
	Videos+depths TR	17.04%	16.36%	16.64%	14.43%	15.83%	16.78%	9.31%	11.59%	**14.75%**
	Accuracy	90.60%	89.84%	87.56%	87.68%	83.11%	88.39%	88.47%	89.48%	**88.14%**
Complete version of online-learning based hybrid complexity reduction method	Video 1 (DVt) TR	**68.34%**	**66.30%**	**66.70%**	**71.44%**	**68.71%**	**68.72%**	**67.12%**	**64.28%**	**67.70%**
	Videos TR	**44.02%**	**49.83%**	**43.35%**	**40.86%**	**44.84%**	**44.01%**	**43.72%**	**40.61%**	**43.90%**
	Videos+depths TR	**23.61%**	**24.97%**	**23.64%**	**25.77%**	**24.09%**	**25.00%**	**17.28%**	**19.70%**	**23.01%**
	Accuracy	**96.11%**	**95.38%**	**96.61%**	**97.09%**	**96.89%**	**96.85%**	**96.69%**	**96.91%**	**96.57%**

are able to fine-tune the model. Regarding DV_t time reduction, using the Laplace smoothing results in performance degradation of 1.75% compared to the complete version of the online-learning hybrid scheme (see Table 4.5). That is because at the beginning of the fine-tuning process, the FMAs are not able to suggest the mode candidates.

In the case that in the online-learning hybrid complexity reduction scheme, when the RD cost threshold conditions are disabled, the DV_t mode prediction accuracy drops by 1.1% (see Table 4.5). This is because in the complete version of the online-learning hybrid scheme, the RD cost threshold is utilized in addition to the probability conditions. Using the RD cost threshold reduces the chance of selecting wrong modes and overcomes the problem of over-fitting to the training data described in Section 4.3.1. Using the RD cost threshold is important, especially when a scene change occurs and at the beginning of encoding a new video sequence. Note that, in the case that the RD cost threshold is disabled, the average DV_t time reduction is 66.48% while the average DV_t mode accuracy is 95.47%. When the probability conditions are not checked in the online-learning based hybrid scheme, the DV_t mode prediction accuracy and DV_t execution time reduction percentage are on average 8.43% and 22.56% lower than those of the complete version of the online-learning hybrid scheme (see Table 4.5). These results show the importance of using the probability conditions.

In Table 4.5, in addition to the video 1 (DV_t) time reduction percentage, we also report the BV_t+DV_t (videos) and videos+depths (BV_t+depth0+DV_t+depth1) encoding execution time reduction percentage for each method. In our study, we used the Bugaboo Dell Xeon cluster from WestGrid, a supercomputing platform in Western Canada. A blade with an Intel Xeon X5650 6-core processor, running at 2.66GHz, and 8-GB RAM was used for the simulations. As we see in Table 4.5, the BV_t+DV_t and videos+depths execution time reduction percentage

achieved by the complete version of the online-learning based hybrid scheme reach 49.83% and 25.77%, respectively, compared to unmodified 3D-HTM.

In summary, the results show the superiority of the complete version of the online-leaning hybrid complexity reduction method over the state-of-the-art methods.

4.5 Conclusions

In this Chapter, we proposed a content dependent complexity reduction for 3D-HEVC. Our method adaptively adjusts the motion search range and introduces an early termination for inter/intra prediction mode search by taking into consideration the disparity between the base view and the other (dependent) views. Performance evaluations show that for the same quality our approach reduces the computational complexity of 3D-HEVC by 29.27% for the dependent texture view encoding time and by 9.71% for the videos+depth maps encoding time, on average.

Then, we proposed a fast mode decision scheme for 3D-HEVC to improve the efficiency achieved by our content dependent complexity reduction. Our method uses a Bayesian classifier to predict the block mode in the dependent view using information of already encoded neighboring blocks in the base and dependent views. Performance evaluations show that our approach significantly reduces the coding complexity of 3D-HEVC (up to 41.96% for the dependent texture view) while minimally hampering the overall bitrate.

In order to further improve the performance of Bayesian based fast mode decision scheme, this Chapter presents an efficient complexity reduction scheme for the encoding process of the dependent texture views of the 3D-HEVC encoder. In this regard, we introduce two new fast mode assignments (FMAs) to predict the mode with the lowest rate distortion cost for the coding units (CUs) in the dependent texture views. The first FMA builds a probabilistic model that uses

the mode information of the CUs belonging to the same quad-tree structure with in one coding tree unit to predict the mode of to-be-coded CU in the current dependent texture view. For mode prediction, the first FMA, also considers if the CUs belonging to the same quad-tree structure with in one coding tree unit are all-zero blocks (AZBs). The second FMA builds a probabilistic model that uses the mode information, AZB information, and motion homogeneity of already encoded CUs in the base texture view and the current dependent texture view to predict the mode of to-be-coded CU in current dependent texture view. Finally, we proposed an online-learning based hybrid complexity reduction scheme based on all the proposed ideas. Our hybrid method adaptively adjusts the motion search range and decreases the complexity of inter/intra prediction mode search. During the encoding process, online-learning is used to fine-tune the two FMA probabilistic models, using the feedback it receives from the encoder. We examine our hybrid approach for the case that there are two texture views and their corresponding depth maps. Performance evaluations show that our hybrid approach outperforms other state-of-the-art schemes by significantly reducing the execution time of the encoder (67.70% for the dependent texture view encoding time and by 23.01% for the videos+depth maps encoding time, on average), while minimally hampering the overall bitrate/quality.

5. Conclusions and Future Work

5.1 Significance and Potential Applications of the Research

Scalable coding enables single channel transmission, which can serve many different quality levels and resolutions. This allows for cost effective universal access of digital media by a variety of playback devices and different bandwidth requirements (ranging from TV displays and computers to tablets and smart phones), an attractive proposition for broadcasters and end-users alike. In this work we presented coding methods that significantly reduce the complexity of the original scalable standards, leading to simpler hardware and software implementations, which have the potential to accelerate the wide adoption of these standards. Our methods are implemented on MPEG's SHVC reference software model (SHM) and 3D-HEVC reference software model (3D-HTM). As such, our contributions are ready to be used by the industry. Note that all our methods are introduced at the encoder side with the decoder left untouched. The performance of our methods is compared with that of the unmodified reference codecs in terms of execution time and compression performance, as recommended by MPEG. Related contributions were submitted and presented at the ITU/MPEG meetings.

In particular, we develop two complexity reduction schemes for spatial SHVC (Chapter 2) and four complexity reduction schemes for quality SHVC (Chapter 3). Although the methods developed for SHVC and presented in these Chapters were designed and tested for standard dynamic range (SDR) content, they may also be used for High dynamic range (HDR) applications [116]. The latter is the emerging "revolution" in digital media and scalability may be considered as the means for supporting compatibility with standard dynamic range content and displays. In this scenario we could consider the transmission of SDR and HDR content of the

same video in one bit-stream. In this case, the SDR content is encoded as the BL and its HDR equivalent is encoded as the EL using SHVC [117], [118]. Another scenario may involve transmission of several different quality and resolution versions of the same HDR video in one bit-stream. In both scenarios, all our proposed methods for quality and spatial SHVC may be used with adjustments/modifications that address HDR content. Details for these changes are described in the Section for "Future Work".

In Chapter 4, we proposed several complexity reduction methods for 3D-HEVC, the latest Multiview/3D compression standard. Lately, TV manufacturers are placing their hopes for the future of 3D TV on the so-called "glasses-free" 3D TV technology, where "autostereoscopic" displays show multiple views of 3D (ranging from 8 to more than 100) without the need for glasses. Autostereoscopic glasses-free displays and 360^0 video applications, which are just emerging require a large number of views that translate to large amounts of data and of course challenging issues in relation to transmission and storage. Our proposed methods were utilized to reduce complexity reduction for the case of having 3D streams with 2 views and their corresponding depth maps, but they can be easily adopted for the autostereoscopic and 360^0 video applications. Another attractive application for these methods is the case of 3D virtual reality and augmented reality [119] where encoding delays for real-time applications will be a challenge. In such applications, for instance, any movement of the user wearing augmented reality glasses in the center of a scene, will require the transmission of a new video scene to the glasses.

News and events transmitted from remote scenes to the main station by reporters and crews using portable cameras also face power consumption challenges. Our methods may directly address these challenges in cases of scalable streaming and multiview broadcasting

5.2 Summary of Contributions

The research presented in this thesis aims at reducing the computational complexity of scalable coding. The methods presented in this thesis use the correlations that exist between the different scalable layers, to reduce the complexity of scalable video coding. In order to extract these correlations, we use statistical studies, regression, and machine learning probabilistic approaches to design content adaptive complexity reduction methods.

This Thesis addresses three complexity reduction topics related to scalable coding. More specifically, we 1) developed two content adaptive complexity reduction methods for the spatial extension of HEVC, 2) designed four content adaptive complexity reduction schemes for the quality extension of HEVC, and 3) proposed three content adaptive complexity reduction methods for 3D-HEVC.

- We propose an adaptive search range adjustment for the motion estimation of spatial SHVC using statistical studies to reduce the number of search points in motion estimation. Our proposed method uses the motion information of the BL to predict appropriate search range value for the EL. Instead of using large fixed search range our method adaptively sets the search range value, which results in significant complexity reduction.
- We also build an EL quad-tree partitioning prediction method at the CU level using Bayesian approach for the spatial SHVC. The method uses a novel CTU labeling system, which makes the training process possible. Our method significantly reduces the computational complexity of the SHVC encoder by reducing the number of partitioning structures that the encoder checks during the course of encoding.

- Next we combine the adaptive search and quad-tree prediction methods into one complexity reduction scheme for spatial scalability for SHVC The combined method outperforms every other existing method.
- We propose an adaptive search range adjustment for the quality scalable extension of HEVC to reduce the number of search points in motion estimation. The method classifies each BL CTU based on motion homogeneity and suggests appropriate search range for the corresponding EL CTU. As a result, it significantly reduces the complexity of encoding EL in quality SHVC.
- We propose an EL early termination method for the exhaustive mode search process of quality SHVC. The method predicts a threshold for the RD cost of each EL block and terminates the mode search process accordingly. This method reduces the number of modes that is checked for each EL block and as a result it significantly reduces the complexity of quality SHVC.
- We design a content adaptive complexity reduction scheme which uses the adaptive search range adjustment method proposed for quality SHVC and the early termination method to achieve higher complexity reduction.
- In order to further reduce the complexity of the search mode, we design three mode prediction schemes for quality scalability. First, we use statistical analysis to propose a hybrid complexity reduction scheme for mode prediction. This method significantly improves the complexity cost but it also increases the bit rate.
- To address the above drawback, we propose an EL fast mode assignment (FMA) for quality SHVC based on the Naive Bayes approach. This method uses a set of

probabilistic models and the encoding information of the early frames of each scene to reduce prediction error.

- We propose an online-learning based inter/intra mode prediction method, which is an extension of the Bayesian method. This method gradually fine-tunes its model during the course of encoding, avoiding the need for using early frames.
- In order to achieve higher performance for quality SHVC, we combine our best mode prediction method (online-learning based FMA) with our content adaptive complexity reduction scheme. The combined method reduces the complexity of motion estimation and exhaustive inter/intra mode search process of the EL in quality SHVC, outperforming all our individual schemes and every other existing method.
- We propose a content adaptive search range adjustment and early termination method for exhaustive mode search for 3D-HEVC.
- We design a Bayesian based mode prediction for the exhaustive mode search process of 3D-HEVC to further improve the performance of our content adaptive method.
- Finally, we propose an online-learning based complexity reduction method that incorporates the two above-mentioned methods to achieve the highest complexity reduction. Using the online learning approach improves the prediction accuracy and augments the achieved complexity reduction performance.

5.3 Directions for Future Work

In Chapters 2 and 3, a Bayesian based quad-tree prediction method at CU level is proposed for spatial and quality SHVC. However, this method is not able to predict the quad-tree partitioning structure at PU and TU level. One direction for future research would be to further

reduce the complexity of the SHVC quad-tree partitioning process by designing a quad-tree prediction method that is able to predict CTU partitioning structures at the PU and TU level in addition to the CU level. In this regard, another machine learning approach and different labeling systems may need to be utilized.

Another future research topic of interest is to design mode prediction methods for spatial SHVC. In this regard, the FMA method based on the Naive Bayes approach designed for quality scalability can be utilized. The general concept of online-learning based mode prediction approach proposed for quality scalability (Section 3.2.3) can be used for spatial scalability. However, the RD cost threshold prediction approach which is used by the online-learning based mode prediction method needs to be modified. This is due to the fact that it is designed for the quality scalability using the RD cost information of the BL and EL. In the case of spatial scalability, corresponding block in the BL is expanded to more than one block (of the same size of that of the BL) in the EL. As a result, the RD cost prediction equation (3.5) in the case of spatial scalability becomes equal to weighting average of the RD cost values of the neighbouring EL blocks and independent of RD cost values of BL blocks (for the spatial ratio of 2). However, the weighting constants in RD cost prediction approach were computed using the RD cost values of the BL blocks as well as those of EL. Thus, the weighting constants are not valid for spatial scalability. Since the RD cost threshold affects the complexity reduction performance and mode prediction accuracy of online-learning based mode prediction approach method, the RD cost threshold prediction method needs to be accurately modified for spatial scalability. To address this issue, currently we are working on a scheme, which predicts the RD cost in the case of spatial scalability. By using this RD cost prediction method, the online-learning mode prediction approach can be utilized for spatial SHVC.

Once we address all the above issues, an important step will be to combine the spatial and quality methods into one scheme and test its overall performance.

In Chapter 4, three methods are proposed for reducing the complexity of the exhaustive mode search process and motion estimation of 3D-HEVC for dependent texture views. However, these methods are not able to reduce the complexity of quad-tree partitioning of 3D-HEVC. Therefore, in order to further reduce the complexity of 3D-HEVC's encoder, a new method could be developed to predict the quad-tree partitioning structure of 3D-HEVC. The Bayesian based quad-tree prediction method (at CU level) that is proposed for SHVC needs to be verified and tested for 3D-HEVC.

The depth map coding costs about 17% of the total encoding time of 3D-HEVC in the case of 2views+ corresponding depth maps, on average. Thus, a related direction for research could be to work on complexity reduction of depth map coding of 3D-HEVC. The existing complexity reduction methods proposed for 3D-HEVC's depth map coding [76], [77], do not address the complexity of mode search process and CTU partitioning. Therefore, new methods are needed to address the complexity reduction of the two procedures of depth map coding in 3D-HEVC. These methods can be designed based on statistical studies and machine learning approaches.

Another direction for future work would be to develop complexity reduction methods for 2D and 3D HDR content. The mode decision making process and quad-tree partitioning of HEVC based codecs (SHVC and 3D-HEVC) depends on the content and the encoding configuration. In order to make the adaptive search range methods proposed for SHVC and 3D-HEVC work for HDR content, statistical studies need to be conducted on HDR training videos to find the relation between the motion information of layers/views. In the early termination mode search methods proposed in Section 3.1.2 and 4.1.2, the weighting constants used for the RD cost prediction

were calculated using SDR content. Therefore, they need to be recalculated using the HDR training video sequences. As mentioned in Chapter 3, the hybrid complexity reduction method proposed for quality SHVC is designed based on statistical studies which were conducted on 2D SDR training videos. Therefore, these results may not be valid for the case of 2D HDR contents. New statistical studies need to be conducted for 2D HDR content. The Bayesian based methods proposed for SHVC and 3D-HEVC use models created using SDR training videos. In the case of HDR, new models have to be generated using HDR content. As it is mentioned in Section 3.2.3, the online learning based mode prediction method proposed for quality SHVC uses the RD cost prediction approach proposed for the early termination. In order to make it work for HDR content, the weighting constants of RD cost prediction approach need to be recalculated as mentioned above. Our online-learning based hybrid complexity reduction scheme, which was proposed for 3D-HEVC, uses the adaptive search range method and early termination method proposed in Section 4.1. These methods need to be modified using HDR content.

Another interesting direction for future research is light field compression. Light fields can be reconstructed from images captured by the different microlenses used in Light Field cameras. In contrast with a conventional camera, which records only *light* intensity, light field technology includes the intensity of *light* in a scene, and also the direction that the *light* rays are traveling in space. Every pixel has color properties, directional properties, and its exact placement in space. Light field technology introduces much more data (light intensity and direction at the same time) and a new media format, demanding new and much more efficient processing schemes than the ones currently used and different network requirements [120]. It is imperative to try to use our knowledge from scalable video coding to try to design new compression schemes for light field video content.

Recently, Joint Video Exploration Team (JVET) of ITU-T has started working on the advanced version of HEVC [119] known as the "Future Generation of Video Coding". Expanding our research to apply to the scalable version of this new codec is only natural.

Bibliography

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[2] K.-C Yang, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard." [Online]. Available: vc.cs.nthu.edu.tw/courses/ISA526100/slides/SVC(project).ppt. [Accessed: 29-Apr-2016].

[3] ITU-T WP3/16 and ISO/IEC JTC1/SC29/WG11, "Joint Preliminary Call for Proposals on Scalable Video Coding Extensions of High Efficiency Video Coding (HEVC)." Doc. w12784, May-2012.

[4] M. T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "HEVC: The New Gold Standard for Video Compression: How Does HEVC Compare with H.264/AVC," IEEE Consum. Electron. Mag., vol. 1, no. 3, pp. 36–46, Jul. 2012.

[5] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[6] M. Hannuksela, K. Ugur, J. Lainema, D. Rusanovskyy, J. Chen, V. Seregin, Y. Wang, Y. Chen, L. Guo, M. Karczewicz, Y. Ye, and J. Boyce, "Test Model for Scalable Extensions of High Efficiency Video Coding (HEVC)." JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCTVC-L0453, Geneva, Jan-2013.

[7] H. R. Tohidypour, M. T. Pourazad, P. Nasiopoulos, and J. Slevinsky, "A new mode for coding residual in scalable HEVC (SHVC)," in 2015 IEEE International Conference on Consumer Electronics (ICCE), 2015, pp. 372–373.

[8] G. Tech, Y. Chen, K. Müller, J. R. Ohm, A. Vetro, and Y. K. Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 35–49, Jan. 2016.

[9] "Test Model 10 of 3D-HEVC and MV-HEVC." Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-J1003, Strasbourg, Oct-2014.

[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

[11] G. J. Sullivan, J.-R. Ohm, F. Bossen, and T. Wiegand and Jizheng Xu, "JCT-VC AHG report: HM subjective quality investigation (AHG22)." JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-H0022r1, San José, CA, Feb-2012.

[12] V. Sze, M. Budagavi, and G. J. Sullivan, Eds., High Efficiency Video Coding (HEVC): Algorithms and Architectures. Cham: Springer International Publishing, 2014.

[13] M. Wien, High Efficiency Video Coding: Coding Tools and Specification. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[14] K. McCann, B. Bross, W.-J. Han, I. K. Kim, K. Sugimoto, and G. J. Sullivan, "HM9: High Efficiency Video Coding (HEVC) Test Model 9 Encoder Description." JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-K1002v2, Shanghai, CN, Oct-2012.

[15] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards #x2014;Including High Efficiency Video Coding (HEVC)," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[16] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[17] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1885–1898, Dec. 2012.

[18] H. Zhang and Z. Ma, "Fast Intra Mode Decision for High Efficiency Video Coding (HEVC)," IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 4, pp. 660–668, Apr. 2014.

[19] S. Cho and M. Kim, "Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 9, pp. 1555–1564, Sep. 2013.

[20] H. Sun, D. Zhou, and S. Goto, "A Low-Complexity HEVC Intra Prediction Algorithm Based on Level and Mode Filtering," in 2012 IEEE International Conference on Multimedia and Expo, 2012, pp. 1085–1090.

[21] Y. Liu, X. Liu, and P. Wang, "A Texture Complexity Based Fast Prediction Unit Size Selection Algorithm for HEVC Intra-coding," in 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE), 2014, pp. 1585–1588.

[22] Y. Zhang, S. Kwong, G. Zhang, Z. Pan, H. Yuan, and G. Jiang, "Low Complexity HEVC INTRA Coding for High-Quality Mobile Video Communication," IEEE Trans. Ind. Inform., vol. 11, no. 6, pp. 1492–1504, Dec. 2015.

[23] N. Dhollande, O. L. Meur, and C. Guillemot, "HEVC Intra coding of ultra HD video with reduced complexity," in 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 4122–4126.

[24] W. Geuder, P. Amon, and E. Steinbach, "Low-complexity block size decision for HEVC intra coding using binary image feature descriptors," in 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 242–246.

[25] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, “Efficient coding unit size selection based on texture analysis for HEVC intra prediction,” in 2014 IEEE International Conference on Multimedia and Expo (ICME), 2014, pp. 1–6.

[26] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, “An Effective CU Size Decision Method for HEVC Encoders,” IEEE Trans. Multimed., vol. 15, no. 2, pp. 465–470, Feb. 2013.

[27] J. Kim, J. Yang, K. Won, and B. Jeon, “Early determination of mode decision for HEVC,” in Picture Coding Symposium (PCS), 2012, 2012, pp. 449–452.

[28] Z. Pan, S. Kwong, M. T. Sun, and J. Lei, “Early MERGE Mode Decision Based on Motion Estimation and Hierarchical Depth Correlation for HEVC,” IEEE Trans. Broadcast., vol. 60, no. 2, pp. 405–412, Jun. 2014.

[29] N. Hu and E. H. Yang, “Fast Motion Estimation Based on Confidence Interval,” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 8, pp. 1310–1322, Aug. 2014.

[30] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, “Early SKIP Detection for HEVC.” JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-G543, Geneva, Nov-2011.

[31] J. Xiong, H. Li, Q. Wu, and F. Meng, “A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence,” IEEE Trans. Multimed., vol. 16, no. 2, pp. 559–564, Feb. 2014.

[32] G. Correa, P. Assuncao, L. Agostini, and L. A. D. S. Cruz, “Coding Tree Depth Estimation for Complexity Reduction of HEVC,” in Data Compression Conference (DCC), 2013, pp. 43–52.

[33] X. Yang, G. Teng, H. Zhao, G. Li, P. An, and G. Wang, “Fast PU decision algorithm based on texture complexity in HEVC,” in 2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 2014, pp. 321–325.

[34] H. S. Kim and R. H. Park, “Fast CU Partitioning Algorithm for HEVC Using an Online-Learning-Based Bayesian Decision Rule,” IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 130–138, Jan. 2016.

[35] M. Li, K. Chono, and S. Goto, “Low-complexity merge candidate decision for fast HEVC encoding,” in 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2013, pp. 1–6.

[36] S. Kim, C. Park, H. Chun, and J. Kim, “A novel fast and low-complexity Motion Estimation for UHD HEVC,” in Picture Coding Symposium (PCS), 2013, 2013, pp. 105–108.

[37] G. Correa, P. Assuncao, L. A. da S. Cruz, and L. Agostini, “Computational complexity control for HEVC based on coding tree spatio-temporal correlation,” in 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), 2013, pp. 937–940.

[38] T. Sotetsumoto, T. Song, and T. Shimamoto, “Low complexity algorithm for sub-pixel motion estimation of HEVC,” in 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC), 2013, pp. 1–4.

[39] H. Maich, V. Afonso, B. Zatt, L. Agostini, and M. Porto, “HEVC Fractional Motion Estimation complexity reduction for real-time applications,” in 2014 IEEE 5th Latin American Symposium on Circuits and Systems (LASCAS), 2014, pp. 1–4.

[40] K. Miyazawa, T. Murakami, A. Minezawa, and H. Sakate, “Complexity reduction of in-loop filtering for compressed image restoration in HEVC,” in Picture Coding Symposium (PCS), 2012, 2012, pp. 413–416.

[41] R. Adireddy and N. K. Palanisamy, “Effective approach to reduce complexity for HEVC intra prediction in inter frames,” in 2014 Twentieth National Conference on Communications (NCC), 2014, pp. 1–5.

[42] M. Naccari, C. Brites, J. Ascenso, and F. Pereira, “Low complexity deblocking filter perceptual optimization for the HEVC codec,” in 2011 18th IEEE International Conference on Image Processing, 2011, pp. 737–740.

[43] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, “Effective coding unit size decision based on motion homogeneity classification for HEVC inter prediction,” in 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 3691–3695.

[44] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, “Fast coding unit size selection for HEVC inter prediction,” in 2015 IEEE International Conference on Consumer Electronics (ICCE), 2015, pp. 457–458.

[45] C. H. Yeh, K. J. Fan, M. J. Chen, and G. L. Li, “Fast Mode Decision Algorithm for Scalable Video Coding Using Bayesian Theorem Detection and Markov Process,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 4, pp. 563–574, Apr. 2010.

[46] L. Shen and Z. Zhang, “Content-Adaptive Motion Estimation Algorithm for Coarse-Grain SVC,” IEEE Trans. Image Process., vol. 21, no. 5, pp. 2582–2591, May 2012.

[47] S. W. Jung, S. J. Baek, C. S. Park, and S. J. Ko, “Fast Mode Decision Using All-Zero Block Detection for Fidelity and Spatial Scalable Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 2, pp. 201–206, Feb. 2010.

[48] S. Lim, J. Yang, and B. Jeon, “Fast Coding Mode Decision for Scalable Video Coding,” in 10th International Conference on Advanced Communication Technology, 2008. ICACT 2008, 2008, vol. 3, pp. 1897–1900.

[49] H. C. Lin, W. H. Peng, H. M. Hang, and W. J. Ho, “Layer-Adaptive Mode Decision and Motion Search for Scalable Video Coding with Combined Coarse Granular Scalability (CGS) and Temporal Scalability,” in 2007 IEEE International Conference on Image Processing, 2007, vol. 2, p. II-289-II-292.

[50] L. Shen, Z. Liu, P. An, R. Ma, and Z. Zhang, “Fast mode decision for scalable video coding utilizing spatial and interlayer correlation,” J. Electron. Imaging, vol. 19, no. 3, pp. 33010-33010–8, 2010.

[51] C. S. Park, B. K. Dan, H. Choi, and S. J. Ko, “A Statistical Approach for Fast Mode Decision in Scalable Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 12, pp. 1915–1920, Dec. 2009.

[52] L. Shen, Y. Sun, Z. Liu, and Z. Zhang, “Efficient SKIP Mode Detection for Coarse Grain Quality Scalable Video Coding,” IEEE Signal Process. Lett., vol. 17, no. 10, pp. 887–890, Oct. 2010.

[53] S. T. Kim, K. R. Konda, P. S. Mah, and S. J. Ko, “Adaptive mode decision algorithm for inter layer coding in scalable video coding,” in 2010 IEEE International Conference on Image Processing, 2010, pp. 1297–1300.

[54] L. Shen, Z. Liu, P. An, R. Ma, and Z. Zhang, “An adaptive early termination of mode decision using inter-layer correlation in scalable video coding,” in 2010 IEEE International Conference on Image Processing, 2010, pp. 4229–4232.

[55] A. Huang, X. Lin, and Y. Chen, “Fast Mode Decision Algorithm for Spatial and Coarse Grain Quality Scalable Video Coding,” in International Symposium on Computer Network and Multimedia Technology, 2009. CNMT 2009, 2009, pp. 1–4.

[56] B. Lee and M. Kim, “A Low Complexity Mode Decision Method for Spatial Scalability Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 1, pp. 88–95, Jan. 2011.

[57] H. Li, Z. G. Li, and C. Wen, “Fast Mode Decision Algorithm for Inter-Frame Coding in Fully Scalable Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 7, pp. 889–895, Jul. 2006.

[58] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, “SHVC Test Model 10 (SHM 10) Introduction and Encoder Description.” JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCTVC-U1007, Warsaw, PL, Jun-2015.

[59] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro, “Standardized Extensions of High Efficiency Video Coding (HEVC),” IEEE J. Sel. Top. Signal Process., vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[60] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, “Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding Standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 20–34, Jan. 2016.

[61] S. Lasserre, F. L. Léannec, J. Taquet, and E. Nassor, “Low-Complexity Intra Coding for Scalable Extension of HEVC Based on Content Statistics,” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 8, pp. 1375–1389, Aug. 2014.

[62] D. Wang, C. Yuan, Y. Sun, J. Zhang, and H. Zhou, “Fast Mode and Depth Decision Algorithm for Intra Prediction of Quality SHVC,” in Intelligent Computing Theory, D.-S.

Huang, V. Bevilacqua, and P. Premaratne, Eds. Springer International Publishing, 2014, pp. 693–699.

[63] X. Zuo and L. Yu, “Fast mode decision method for all intra spatial scalability in SHVC,” in 2014 IEEE Visual Communications and Image Processing Conference, 2014, pp. 394–397.

[64] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y.-K. Wang, “Joint draft 8.0 on multiview video coding.” Joint Video Team, Doc. JVT-AB204, Hannover, Germany, Jul-2008.

[65] L. Shen, Z. Liu, T. Yan, Z. Zhang, and P. An, “View-Adaptive Motion Estimation and Disparity Estimation for Low Complexity Multiview Video Coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 6, pp. 925–930, Jun. 2010.

[66] Q. Zhang, P. An, Y. Zhang, L. Shen, and Z. Zhang, “Low complexity multiview video plus depth coding,” IEEE Trans. Consum. Electron., vol. 57, no. 4, pp. 1857–1865, Nov. 2011.

[67] L. Shen, Z. Liu, S. Liu, Z. Zhang, and P. An, “Selective Disparity Estimation and Variable Size Motion Estimation Based on Motion Homogeneity for Multi-View Coding,” IEEE Trans. Broadcast., vol. 55, no. 4, pp. 761–766, Dec. 2009.

[68] L. Shen, Z. Liu, P. An, R. Ma, and Z. Zhang, “Low-Complexity Mode Decision for MVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 6, pp. 837–843, Jun. 2011.

[69] Z. P. Deng, Y. L. Chan, K. B. Jia, C. H. Fu, and W. C. Siu, “Fast Motion and Disparity Estimation With Adaptive Search Range Adjustment in Stereoscopic Video Coding,” IEEE Trans. Broadcast., vol. 58, no. 1, pp. 24–33, Mar. 2012.

[70] Y. Kim, J. Kim, and K. Sohn, “Fast Disparity and Motion Estimation for Multi-view Video Coding,” IEEE Trans. Consum. Electron., vol. 53, no. 2, pp. 712–719, May 2007.

[71] H. R. Tohidypour, M. T. Pourazad, P. Nasiopoulos, and V. Leung, “A content adaptive complexity reduction scheme for HEVC-based 3D video coding,” in 2013 18th International Conference on Digital Signal Processing (DSP), 2013, pp. 1–5.

[72] H. Schwarz et al., “3D video coding using advanced prediction, depth modeling, and encoder control methods,” in Picture Coding Symposium (PCS), 2012, 2012, pp. 1–4.

[73] S. Ma, S. Wang, and W. Gao, “Low Complexity Adaptive View Synthesis Optimization in HEVC Based 3D Video Coding,” IEEE Trans. Multimed., vol. 16, no. 1, pp. 266–271, Jan. 2014.

[74] J. W. Kang, Y. Chen, L. Zhang, and M. Karczewicz, “Low complexity Neighboring Block based Disparity Vector Derivation in 3D-HEVC,” in 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 1921–1924.

[75] E. G. Mora, J. Jung, M. Cagnazzo, and B. Pesquet-Popescu, “Modification of the merge candidate list for dependent views in 3D-HEVC,” in 2013 IEEE International Conference on Image Processing, 2013, pp. 1709–1713.

[76] Q. Zhang, N. Li, Y. Gan, Q. Zhang, N. Li, and Y. Gan, “Low Complexity Mode Decision for 3D-HEVC,” Sci. World J. Sci. World J., vol. 2014, 2014, p. e392505, Aug. 2014.

[77] Q. Zhang, Q. Wu, X. Wang, and Y. Gan, “Early SKIP mode decision for three-dimensional high efficiency video coding using spatial and interview correlations,” J. Electron. Imaging, vol. 23, no. 5, p. 53017, Oct. 2014.

[78] Q. Zhang, H. Chang, Q. Wu, and Y. Gan, “Fast motion and disparity estimation for HEVC based 3D video coding,” Multidimens. Syst. Signal Process., vol. 27, no. 3, pp. 743–761, Nov. 2014.

[79] N. Zhang, D. Zhao, Y.-W. Chen, J.-L. Lin, and W. Gao, “Fast encoder decision for texture coding in 3D-HEVC,” Signal Process. Image Commun., vol. 29, no. 9, pp. 951–961, Oct. 2014.

[80] N. Zhang, Y.W. Chen, J.L. Lin, J. An, K. Zhang, S. Lei, S. Ma, D. Zhao, and W. Gao, “3D-CE3.h related: Fast encoder decision for texture coding.” Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-E0173, Vienna, Jul-2013.

[81] L. Shen, P. An, Z. Zhang, Q. Hu, and Z. Chen, “A 3D-HEVC Fast Mode Decision Algorithm for Real-Time Applications,” ACM Trans Multimed. Comput Commun Appl, vol. 11, no. 3, p. 34:1–34:23, Feb. 2015.

[82] T. Ikai, “CE3-related: Worst case complexity reduction for merge candidate construction.” :JCT3V of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-G0034, San José, Jan-2014.

[83] C.F. Chen, G.G. Lee, and B.S. Li, “CE6-related: Complexity reduction on simplified depth coding (SDC) with subsampling on neighbouring reference pixels.” JCT3V of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-E0127, Vienna, Jul-2013.

[84] G. Sanchez, M. Saldanha, B. Zatt, M. Porto, and L. Agostini, “S-GMOF: A gradient-based complexity reduction algorithm for depth-maps intra prediction on 3D-HEVC,” in 2015 IEEE 6th Latin American Symposium on Circuits Systems (LASCAS), 2015, pp. 1–4.

[85] H. B. Zhang, C. H. Fu, W. M. Su, S. H. Tsang, and Y. L. Chan, “Adaptive fast intra mode decision of depth map coding by Low Complexity RD-Cost in 3D-HEVC,” in 2015 IEEE International Conference on Digital Signal Processing (DSP), 2015, pp. 487–491.

[86] N. Purnachand, L. N. Alves, and A. Navarro, “Fast Motion Estimation Algorithm for HEVC,” in 2012 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2012, pp. 34–37.

[87] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “Adaptive search range method for spatial scalable HEVC,” in 2014 IEEE International Conference on Consumer Electronics (ICCE), 2014, pp. 191–192.

[88] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “Probabilistic Approach for Predicting the Size of Coding Units in the Quad-Tree Structure of the Quality and Spatial Scalable HEVC,” IEEE Trans. Multimed., vol. 18, no. 2, pp. 182–195, Feb. 2016.

[89] K. P. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, 2013.

[90] T. M. Mitchell, “Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression.” [Online]. Available: http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf. [Accessed: 01-May-2016].

[91] C. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[92] “Joint Call for Proposals on Video Compression Technology.” ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, Doc. VCEG-AM91, Kyoto, Jan-2010.

[93] V. Seregin and Y. He, “Common SHM test conditions and software reference configurations.” JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCTVC-Q1009, Valencia, ES, Apr-2014.

[94] J. Chen, J. Boyce, Y. Ye , and M. M. Hannuksela, “Scalable HEVC (SHVC) Test Model 6 (SHM 6).” JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCTVC-Q1007, Valencia, ES, Apr-2014.

[95] “Random access main configuration.” [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-6.1/cfg/encoder_randomaccess_main.cfg. [Accessed: 01-May-2016].

[96] A. Fujibayashi and T.K. Tan, “Random access support for HEVC.” Joint Collaborative Team on Video Coding (JCT-VC), Doc. JCTVC-D234, Daegu, Jan-2011.

[97] “Intra main configuration.” [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-6.1/cfg/encoder_intra_main.cfg. [Accessed: 01-May-2016].

[98] F. Bossen, “Common test conditions and software reference configurations.” JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-L1100, Geneva, CH, Jan-2013.

[99] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves.” ITU-T and VCEG, Doc. VCEG-M33, Austin, Texus, Apr-2011.

[100] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “Content adaptive complexity reduction scheme for quality/fidelity scalable HEVC,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 1744–1748.

[101] H.R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “Content Adaptive Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC.” JCTVC of ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-L0042, Geneva, Jan-2013.

[102] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “An Encoder Complexity Reduction Scheme for Quality/Fidelity Scalable HEVC,” IEEE Trans. Broadcast., vol. 62, no. 3, pp. 664–674, Sep. 2016.

[103] P. Yin, T. Lu, T. Chen, X. Xiu, and Y. Ye, “Non-TE2: Inter-layer reference picture placement.” JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCTVC-L0174, Geneva, CH, Jan-2013.

[104] H.R. Tohidypour, H. Bashashati, M. T. Pourazad, and P. Nasiopoulos, “Fast mode assignment for quality scalable extension of the high efficiency video coding (HEVC) standard: a Bayesian approach,” presented at the Proceedings of the 6th Balkan Conference in Informatics (BCI), 2013, pp. 61–65.

[105] H. R. Tohidypour, H. Bashashati, M. T. Pourazad, and P. Nasiopoulos, “Online-learning Based Mode Prediction Method for Quality Scalable Extension of the High Efficiency Video Coding (HEVC) Standard,” IEEE Trans. Circuits Syst. Video Technol., 2016.

[106] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “A low complexity mode decision approach for HEVC-based 3D video coding using a Bayesian method,” in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 895–899.

[107] H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, “Online Learning-based Complexity Reduction Scheme for 3D-HEVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 10, pp. 1870–1883, Oct. 2016.

[108] M. Tanimoto, T. Fujii, M.P. Tehrani, M. Wildeboer, N. Fukushima, and H. Furihata, “Moving multiview camera test sequences for MPEG-FTV.” ISO/IEC JTC1/SC29/WG11, Doc. M16922, Xian, Oct-2009.

[109] G.M. Um, G. Bang, N. Hur, J. Kim, and Y.S. Ho, “3D video test material of outdoor scene.” ISO/IEC JTC1/SC29/WG11, Doc. M15371, Archamps, Apr-2008.

[110] D. Rusanovskyy, K. Müller, and A. Vetro, “Common Test Conditions of 3DV Core Experiments.” ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-F1100, Geneva, Oct-2013.

[111] J. Zhang, R. Li, H. Li, D. Rusanovskyy, and M. M. Hannuksela, “Ghost Town Fly 3DV sequence for purposes of 3DV standardization.” ISO/IEC JTC1/SC29/WG11, Doc. M20027, Geneva, Switzerland, Mar-2011.

[112] D Rusanovskyy, P Aflaki, MM Hannuksela, “Undo Dancer 3DV sequence for purposes of 3DV standardization.” ISO/IEC JTC1/SC29/WG11, Doc. M20028, Geneva, Mar-2011.

[113] Y.-S. Ho, E.-K. Lee, and C. Lee, “Multiview video test sequence and camera parameters.” ISO/IEC JTC1/SC29/WG11, Doc. m15419, Archamps, France, Apr-2008.

[114] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, and K. Wegner, “Poznań Multiview Video Test Sequences and Camera Parameters.” ISO/IEC JTC1/SC29/WG11, Doc. M17050, Xian, China, Oct-2009.

[115] “Configuration for 3D-HEVC Encoder, Main Profile, 2view+depth.” [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-12.2/cfg/3D-HEVC/baseCfg_2view+depth.cfg. [Accessed: 02-May-2016].

[116] AHG on HDR and WCG, “Draft Call for Evidence (CfE) for HDR and WCG Video Coding (v4).” ISO/IEC JTC1/SC29/WG11, Doc. M35464, Geneva, Switzerland, Feb-2015.

[117] R. Boitard, M. T. Pourazad, P. Nasiopoulos, and J. Slevinsky, “Demystifying High-Dynamic-Range Technology: A new evolution in digital media.,” IEEE Consum. Electron. Mag., vol. 4, no. 4, pp. 72–86, Oct. 2015.

[118] M. Azimi et al., “Compression efficiency of HDR/LDR content,” in Seventh International Workshop on Quality of Multimedia Experience (QoMEX), 2015, pp. 1–6.

[119] J. Ridge and M. M. Hannuksela, “Future video coding requirements on virtual reality.” ISO/IEC JTC1/SC29/WG11 MPEG2016, Doc. M37709, San Diego, US, Feb-2016.

[120] F. Dai, J. Zhang, Y. Ma, and Y. Zhang, “Lenselet image compression scheme based on subaperture images streaming,” in 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 4733–4737.

Appendix

List of Other Publications

[P11] H. R. Tohidypour, M. T. Pourazad, P. Nasiopoulos, and J. Slevinsky, "A new mode for coding residual in scalable HEVC (SHVC)," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, 2015, pp. 372–373.

[P12] M. Azimi, R. Boitard, B. Oztas, S. Ploumis, H. R. Tohidypour, M. T. Pourazad, and P. Nasiopoulos, "Compression efficiency of HDR/LDR content," in *Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, 2015, pp. 1–6.

[P13] H. R. Tohidypour, M. Azimi, M. T. Pourazad, and P. Nasiopoulos, "Software implementation of visual information fidelity (VIF) for HDRTools," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC- W0115, MPEG Doc. 37950, San Diego, US, Feb. 2016.