INVESTIGATION OF SCALABLE HEVC AND ITS BITRATE ALLOCATION FOR UHD DEPLOYMENT IN THE CONTEXT OF HTTP STREAMING

by

UMA SAGAR MADHUGIRI DAYANANDA

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2015

Acknowledgements

I consider it a privilege to express gratitude and respect to all those who guided and helped me throughout the course of my research work.

First and foremost, I take this opportunity to extend gratitude to my thesis advisor, Dr. K.R Rao, for introducing me to this fascinating area of research in video coding, his guidance, support, help rendered and continuous encouragement throughout the research work.

I am thankful to Adobe Systems Inc. for giving me an internship opportunity to carry out the research work. I thank my mentor Dr. Vishy Swaminathan and the Video Systems Team at Adobe Research for guiding and supporting me throughout the work. Vishy has always been there to discuss and provide suggestions.

I extend my thanks to Dr. Michael T. Manry and Dr. William E. Dillon for serving on the thesis committee. I thank my fellow members of Multimedia Processing Lab at UTA for their help and support during the stay. I also thank University of Texas at Arlington for providing necessary resources to carry out my research work.

Finally, I thank my parents, sisters, family members and friends, without whom I would not be who I am today. They always have been the backbone, giving unconditional support and encouragement throughout the journey.

November 25, 2015

Abstract

INVESTIGATION OF SCALABLE HEVC AND ITS BITRATE ALLOCATION FOR UHD DEPLOYMENT IN THE CONTEXT OF HTTP STREAMING

Uma Sagar Madhugiri Dayananda, MS

The University of Texas at Arlington, 2015

Supervising Professor: K. R. Rao

High Efficiency Video Coding (HEVC/H.265) [1] is the state of the art video compression standard, which provides bitrate reduction in the range of 50% when compared to the previous Advanced Video Coding (MPEG-4 AVC/H.264) [5] standard at similar video quality. Scalable High Efficiency Video Coding (SHVC) [2] is the scalable extension of HEVC, which provides traditional scalability options in terms of quality, spatial resolution and temporal frame rate and newer scalability options as well. SHVC can be used to deliver Ultra High Definition (UHD) or 4K resolution video content to mix of clients having varying characteristics.

The distribution of clients in today's multimedia environment is heterogeneous, as there are televisions, computers and mobile devices, supporting different codecs (MPEG-4 AVC or HEVC), resolutions (UHD, HD or SD) and varying bandwidth characteristics. To efficiently deliver video content to heterogeneous clients having varying resources, a combination of these state of the art video coding (SHVC) and streaming technologies (MPEG-DASH) can be employed.

Traditionally, multiple versions of the same video are stored on the servers to satisfy varying client characteristics and are delivered using simulcast coding. This leads to increased video bitrates and hence increases storage costs. However, using scalable video coding such as SHVC - where multiple versions of the video are embedded into different layers of the bit stream, results in bitrate savings. This bitrate savings come at a cost of reduced coding efficiency due to addition of layers, known as scalability overhead.

The primary focus of the thesis is investigation of bitrate savings and the scalability overhead incurred during encoding of UHD video content as SHVC enhancement layer with HEVC or MPEG-4 AVC as HD base layer and obtaining a methodology for comparison of scalable codec such as SHVC with other codecs. Experiments are conducted for SHVC encoding with fixed bitrate allocation into base layer (BL) and enhancement layer (EL) for two and three layers, concentrating on spatial and quality scalabilities. Additional experiments for two layered SHVC encoding are performed by varying the bitrate allocation into BL and EL exploring spatial scalability.

The heuristic method of bitrate allocation for scalable video coding considering both bitrate savings and scalability overhead is a tedious process and error prone. In order to effectively satisfy clients with varying bandwidth characteristics in the context of HTTP video streaming, an optimal SHVC bitrate allocation is necessary. Hence, existing bit rate allocation problem for scalable video coding is reviewed, adapted and evaluated for the scenario of UHD deployment with SHVC for optimal bitrate allocation with two layers.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

INTRODUCTION

1.1 Motivation

Today’s multimedia environment is growing with devices supporting High Definition (HD) and Ultra High Definition (UHD) resolutions, video compression standards such as High Efficiency Video Coding (HEVC/H.265) [1] and Scalable High Efficiency Video Coding (SHVC) [2], and streaming technologies such as Dynamic Adaptive Streaming over HTTP (MPEG-DASH). In order to efficiently deliver Ultra High Definition (UHD) or 4K resolution video content to heterogeneous clients having varying display and bandwidth characteristics, these video coding and streaming technologies can be harnessed. Thus, the research mainly aims at investigating how UHD video content can be delivered to this mix of clients using SHVC.

1.2 Overview

SHVC is the scalable extension to the state of the art video coding standard – HEVC, where the video content can be encoded into base layer and enhancement layer(s) resulting in a single scalable bit stream. This scalable bit stream can be delivered to clients using adaptive HTTP streaming technology. Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH [18], is a streaming technique that enables high quality streaming of multimedia content over Internet to be delivered through conventional HTTP web servers. In HTTP adaptive bitrate video streaming, various versions of the same video are stored on the servers to cater to clients having heterogeneous resources. This increases the storage costs of the Content Delivery Networks (CDNs). A scalable codec such as SHVC which has layers within a single bit stream can be used by different clients to obtain bitrate savings and thus reduce the storage costs. However, this bitrate savings comes at a cost of reduction in coding efficiency known as scalability overhead. In this work, bitrate savings and scalability overhead incurred while encoding UHD video content as enhancement layer of SHVC are investigated.

The existing client distribution is heterogeneous, with few clients supporting only earlier codecs such as MPEG-4 AVC [5] and the other clients supporting HEVC. In order to deliver UHD content to this distribution of clients, SHVC's hybrid codec feature is explored, where the base layer (BL) is coded using MPEG-4 AVC or HEVC. For UHD deployment with SHVC, two deployment scenarios are investigated: (1) Spacial scalability with two layers – BL encoding using MPEG-4 AVC or HEVC for HD resolution, spatial enhancement layer (EL) encoding using SHVC for UHD resolution; (2) Combined spatial and quality scalabilities with 3 layers – BL encoding using MPEG-4 AVC or HEVC for HD resolution, spatial EL1 encoding for UHD resolution and quality EL2 encoding for UHD resolution.

The ratio of allocation of bitrates into base and enhancement layers referred to as bitrate allocation, is varied to satisfy the distribution of clients effectively. Various simulations are conducted to study the bitrate savings and scalability overhead incurred during SHVC encoding for UHD deployment. The existing rate allocation problem is adapted and bitrate allocation strategy is evaluated for two layers in SHVC for UHD deployment.

The main contributions of the thesis are: (1) Methodology to fairly compare scalable codecs with other codecs for simulcast and single layer coding scenarios. (2) Determination of bitrate savings of scalable video coding with SHVC by comparing SHVC with simulcast coding. (3) Determination of scalability overhead of SHVC by comparing with single layer coding. (4) Investigation of optimal allocation of bits into base and enhancement layers of SHVC by varying bit rate allocation ratio in layers of SHVC. (5) Evaluation of rate allocation for SHVC in the context of HTTP streaming for 2 layers.

1.3 Thesis Outline

The thesis is organized as follows: Chapter 2 provides an overview of the High Efficiency Video Coding (HEVC) standard, adaptive bitrate streaming technology such as MPEG-DASH and a brief overview of UHD video format. Chapter 3 presents the overview of Scalable High Efficiency Video Coding (SHVC) standard. Chapter 4 outlines the need for investigation of SHVC, performance comparison metrics and an evaluation methodology. It also describes experimental set-up and the various experiments performed. The results of SHVC investigation and its discussion are provided in Chapter 5. The need for bitrate rate allocation in SHVC, an existing rate allocation problem, its adaptation and evaluation for SHVC in HTTP streaming context are given in Chapter 6. The conclusions and an insight into further work are given in Chapter 7.

Chapter 2

VIDEO CODING AND STREAMING TECHNOLOGIES

2.1 High Efficiency Video Coding

2.1.1 Overview

High Efficiency Video Coding (HEVC) [1] is an international standard for video compression developed by working group of ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). The main goal of HEVC is to significantly improve compression performance compared to existing standards such as H.264/MPEG-4 AVC [5] in the range of 50% bit rate reduction at similar visual quality [1]. It supports resolutions up to 4K and 8K, bit depths of 8, 10, 12 and 16 bits per sample.

HEVC uses block-based hybrid video coding techniques. Redundancies in video sequences can be categorized into spatial, temporal, statistical and perceptual redundancies. Various video coding techniques are used to remove these redundancies in HEVC video codec such as: spatial redundancy removal using intra prediction and block transforms, temporal redundancy removal using inter prediction, statistical redundancy removal using entropy coding and perceptual redundancy using quantization [11].

2.1.2 Encoding and Decoding Workflow

For encoding source video consisting of video frames using HEVC, each frame is partitioned into non-overlapping blocks. A prediction signal is obtained for each of the blocks using intra-prediction or inter-prediction. This prediction signal for a block is subtracted from the original block to obtain residual signal, which is transformed, quantized and entropy encoded into bit-stream. The prediction parameters required to reproduce the signal at the decoder side are also encoded into the bit-stream.

The encoder duplicates the decoder processing loop such that both will generate identical predictions for subsequent data. Therefore, the quantized transform coefficients are constructed by inverse scaling and then inverse transformed to duplicate the decoder approximation of the residual signal. The residual signal is then added to the prediction, and the result is fed into one or two loop filters such as de-blocking filter and sample adaptive offset (SAO) filters to smooth out artifacts induced by the block-wise processing and quantization. The reconstructed picture is stored in the decoded picture buffer which can be used for prediction [1].

The encoder has control engine which decides on the applicable prediction modes, prediction and filtering parameters and also the applicable quantization parameters. Control information about the selected prediction tools and configurations is also included in the bit-stream to inform the decoder [8].

The basic structure of the video encoding scheme based on HEVC is presented in Figure 2-1. The decoder inverses all the operations in encoding and decodes the bit stream. Figure 2-2 represents the decoding scheme based on HEVC.

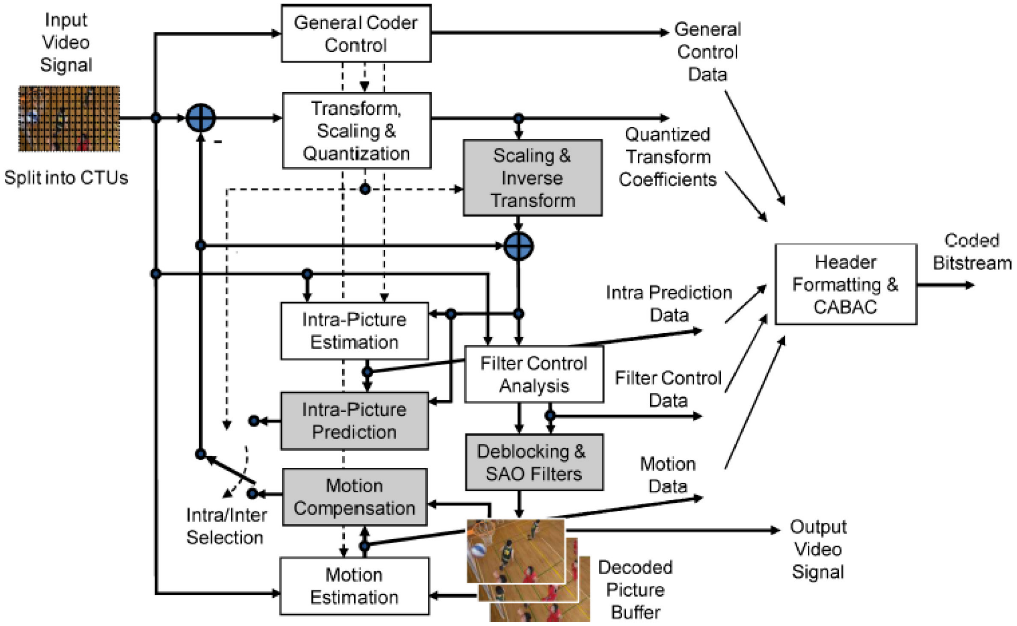


Figure 2-1 Block Diagram of HEVC Encoder (with decoder blocks in shaded grey) [1]

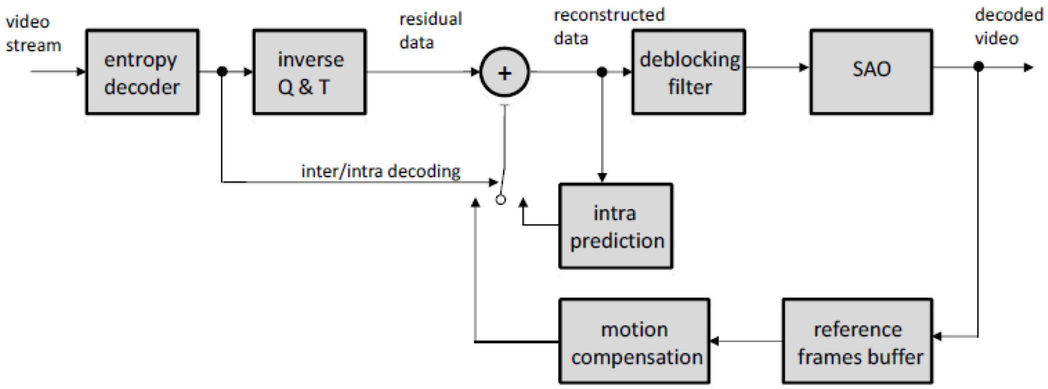


Figure 2-2 Block Diagram of HEVC Decoder [12]

2.1.3 Coding Features

2.1.3.1 Picture Partitioning

In HEVC, each picture is partitioned into square-shaped Coding Tree Blocks (CTBs), where its size varies from 16x16 to 64x64 pixels. One Luma CTB and two chroma CTBs with syntax elements form the Coding Tree Unit (CTU), which is the basic processing unit in HEVC. This is similar to macro-blocks (16x16) in previous H.264/MPEG-4 AVC standard. CTU is sub-divided into square regions called Coding Units (CUs) using quad-tree structure. CU size ranges from 8x8 to 64x64 pixels. Each CU is partitioned into Prediction Units (PUs) which is predicted using intra or inter prediction. The difference of original and prediction in each CU is transformed using one or more block transforms of size varying from 32x32 to 4x4 pixels [13].

The partitioning of picture into CTUs, CU, PU and TU is represented in Figure 2-3 and Figure 2-4 represents partitioning of a video frame of KristenAndSara test sequence using Elecard HEVC Analyzer [61].

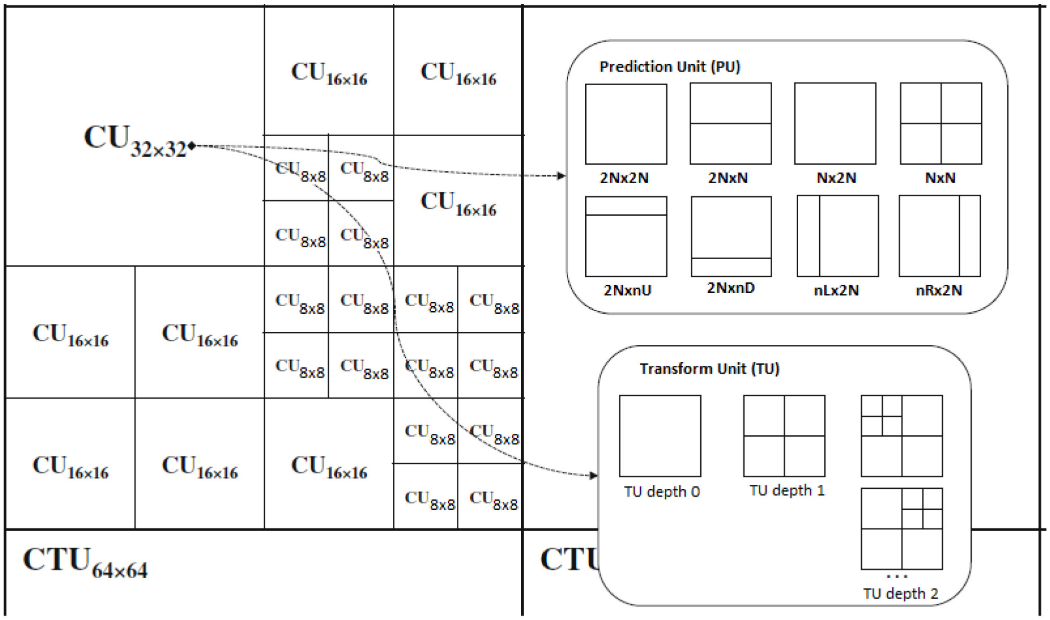
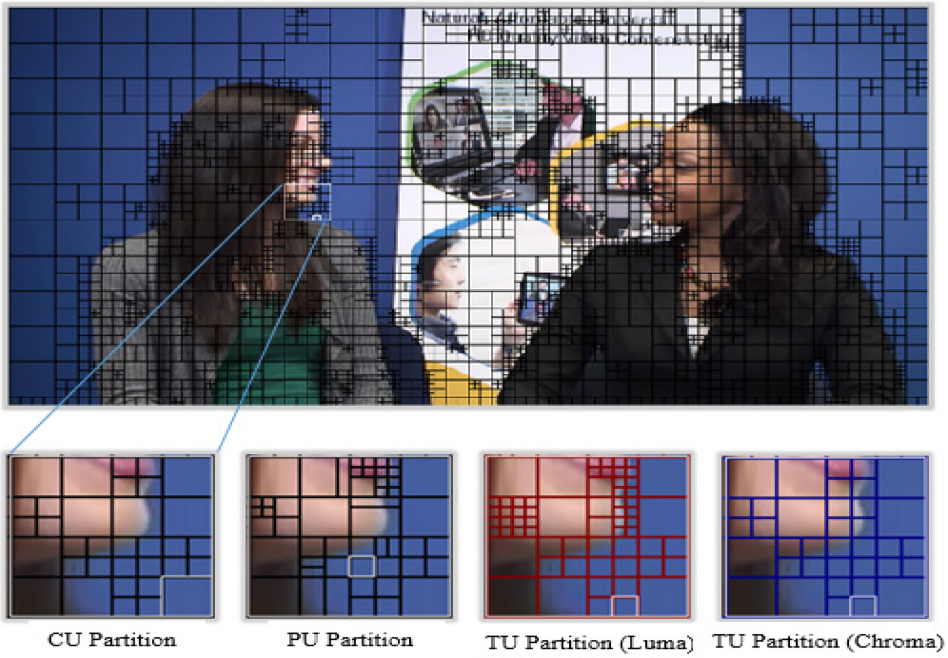


Figure 2-3 Partitioning of picture in HEVC [13]



CU Partition PU Partition TU Partition (Luma) TU Partition (Chroma)

Figure 2-4 Partitioning of a frame in KristenAndSara Test Sequence [61]

2.1.3.2 Prediction schemes

Prediction in HEVC can be intra prediction or inter prediction. The decision whether to apply intra or inter prediction is made at the CU level. A sequence of CTBs is called a slice. CUs in intra mode are predicted from reconstructed neighboring samples within the same slice. In I slice, only intra prediction is enabled for the CUs. In P and B slices, CUs may be in both intra or inter prediction mode. Figure 2-5 represents different prediction modes used in HEVC.

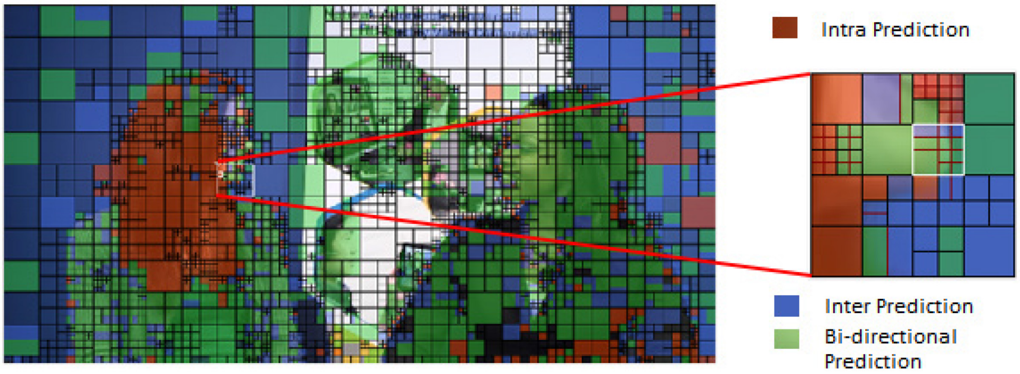


Figure 2-5 Prediction modes in HEVC [61]

Intra Prediction: In this prediction method, blocks are predicted using the neighboring pixels reconstructed from the same frame, exploring the spatial redundancy. For intra-prediction, the basic assumption is that texture of a picture region is similar to the texture in the local neighborhood and hence this is used for prediction. Samples from the top row and from the left column to the current block are used for prediction. The values of the available neighboring samples are combined to form a directional or the planar prediction signal. HEVC has 35 intra prediction modes including a DC, planar and 33 angular prediction modes. The prediction modes 2-18 are the horizontal prediction modes, 19-34 are the vertical prediction modes [8]. Each PU is predicted from neighboring image data in the same picture, using DC prediction (an average value for the PU), planar prediction (fitting a plane surface to the PU) or directional prediction (extrapolating from neighboring data). Intra-coded CUs may only use the partition modes 2Nx2N or NxN, so intra PUs are always square. Figure 2-6 represents the intra prediction modes and directions.

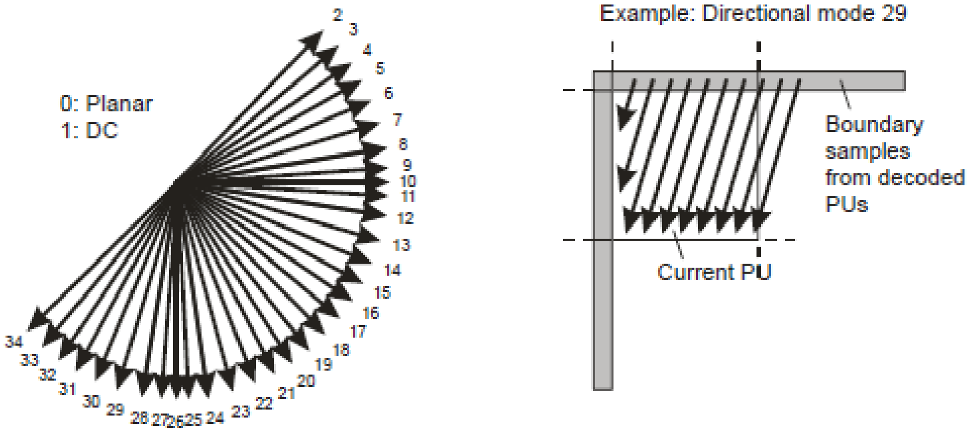


Figure 2-6 Intra prediction modes and prediction directions in HEVC [1]

Inter-prediction: In HEVC, inter prediction is performed at the prediction block (PB) level. It uses temporal redundancy between the adjacent frames in order to predict the current block of frame. Inter prediction is called the motion compensated prediction, since the shifted areas of the reference pictures are used for prediction of the current PB. The resulting displacement between the area in the reference picture and the current PB is interpreted as the motion of the area between the reference picture and the current picture. The encoded displacement or the motion vectors are usually determined by application of the rate-distortion criterion. For inter prediction, the partitions of CB into PB can be symmetric or asymmetric.

Motion compensated prediction can be performed using one or two reference pictures as the prediction source. The number of available prediction sources depends on the slice type to which the PB belongs. For P slices, uni-prediction can be used, which needs only a single prediction reference. For B slices, one or two prediction sources can be applied and it uses two reference picture lists. Bi-prediction uses reference pictures from list 0 and list 1. Here uni-prediction or bi-prediction can be employed. For both uni- and bi-prediction, the construction of PB can be further controlled by applying a configurable weight for each applicable prediction source and an additional offset value, which is called weighted prediction [8]. These weights are explicitly transmitted in the slice header.

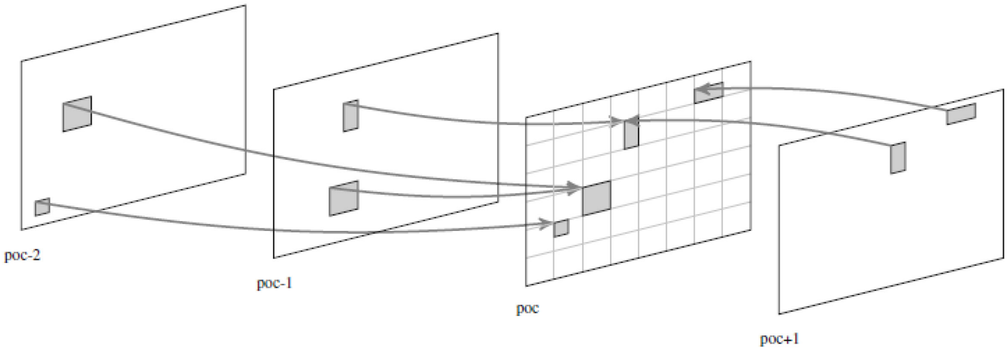
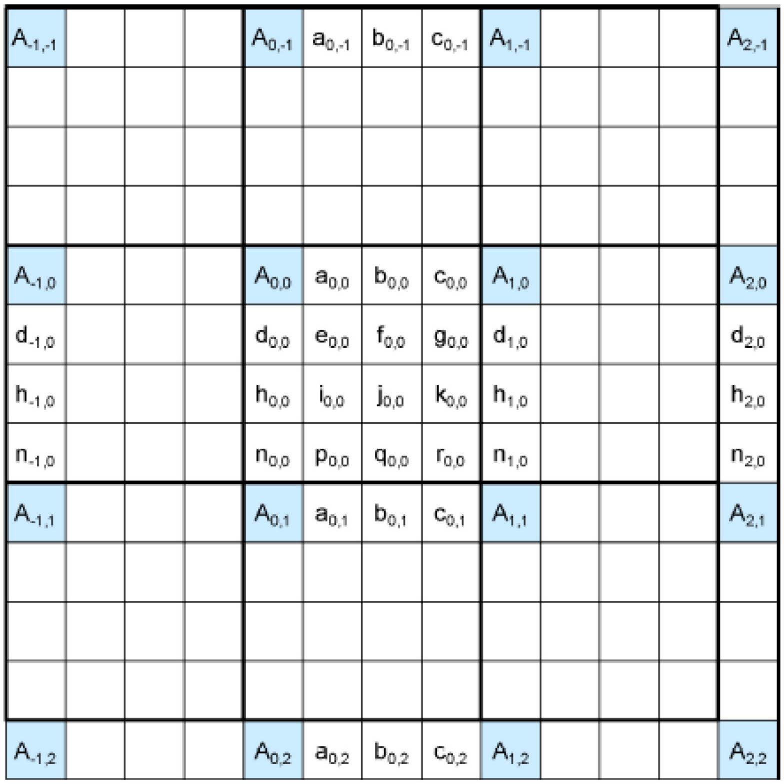


Figure 2-7 Illustration of uni-prediction and bi-prediction [8]

The applicable motion vectors for motion compensation can be derived in two ways - motion vector predictor and motion vector difference. The motion vector predictor is selected from two candidates derived from the spatial and temporal neighborhood of the current block. This predictor selection method is called advanced motion vector prediction. Furthermore, the motion information such as motion vector and reference index can be derived by selection from a configurable set of candidates, without encoding a motion vector difference. The derivation of motion vectors from a candidate set is called merge mode [8].

When the motion vector does not have an integer value, fractional sample interpolation is used to generate the prediction samples for non-integer sampling positions.

HEVC supports motion vectors with units of one quarter of the distance between luma samples and for chroma samples it is dependent on the chroma sampling format. For 4:2:0 sampling format (for every four luma samples, there will be one chroma sample obtained by horizontal and vertical sampling), one eighth of the distance between the chroma samples is used. Fractional interpolation for luma samples uses separable application of an 8-tap filter for the half sample positions and a 7-tap filter for the quarter sample positions. For chroma components, 4-tap filter is used and the fractional accuracy is one eighth for the 4:2:0 chroma format. In Fig. 2-8, the available luma samples at integer sample locations are labeled with upper-case letters, whereas the other positions labeled with lower-case letters represent samples at non-integer sample locations, which need to be generated by interpolation [1].



$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,2}$				$A_{0,2}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,2}$				$A_{2,2}$

Figure 2-8 Integer and fractional sample positions for luma interpolation [1]

2.1.3.3 Transform

The transform process in HEVC is specified using fixed point integer operations with output and intermediate values not exceeding 16-bit word length. HEVC supports four integer transform matrices of sizes 4x4, 8x8, 16x16 and 32x32, which are integer approximations of block transform - Discrete Cosine Transform (DCT - II) [7]. These four integer DCTs are referred to as HEVC core transforms. For intra predicted 4x4 blocks, the discrete sine transform (DST) is used [1].

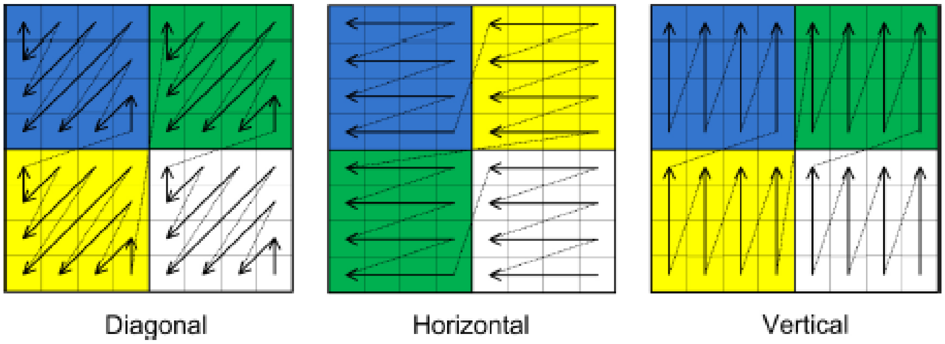


Figure 2-9 Coefficient scanning methods in HEVC [15]

The transforms are separable, where two dimensional transforms are obtained by applying one-dimensional transforms in both the horizontal and vertical directions. Each transform block (TB) is divided into 4x4 sub-blocks (coefficient groups). The processing starts with the last significant coefficient and proceeds to the DC coefficient in the reverse scanning order. The higher precision and larger sizes of the transforms are one of the main reasons HEVC performs better than MPEG-4 AVC [8] [14].

2.1.3.4 Quantization

For quantization, HEVC uses a scalar quantization such as uniform-reconstruction quantization (URQ) scheme controlled by a quantization parameter (QP). Quantization is the main source of loss of information in a lossy video compression such as HEVC. The range of QP values is defined from 0 to 51. The quantizer design in HEVC follows a logarithmic structure with a duplication of the quantizer step size (Δ_q) between QP and QP+6. For an even distribution of the quantizer step size according to this design, the relation between the step sizes is:

$$\Delta_q(QP+1) = \sqrt[6]{2} \cdot \Delta_q(QP) \tag{2.1}$$

Defining $\Delta_q = 1$ for QP = 4, the first six quantizer step sizes are

$$\Delta_{q,0} \in \left\{ 2^{\frac{-4}{6}}, 2^{\frac{-3}{6}}, 2^{\frac{-2}{6}}, 2^{\frac{-1}{6}}, 1, 2^{\frac{1}{6}} \right\} \tag{2.2}$$

The quantizer step sizes for QP > 5 are derived by scaling,

$$\Delta_q(QP) = \Delta_{q,0}(QP \bmod 6) \cdot 2^{\left\lfloor \frac{QP}{6} \right\rfloor} \tag{2.3}$$

With the quantizer range defined as

$$\text{QP} = 0,\ldots,51 \tag{2.4}$$

The quantizer step size is in the range of $0.630 \leq \Delta_q \leq 228.1$. Thereby, the maximum quantizer step is in the range of the maximum amplitude of the 8 bit input signal. For higher bit-depths, a QP offset is specified, increasing the QP parameter range by 6 per additional bit of signal bit depth towards finer quantization [8].

2.1.3.5 Entropy Coding

A coded HEVC bit stream consists of quantized transform coefficients, prediction information such as prediction modes and motion vectors, partitioning information and other header data. All of these elements are encoded using Context Adaptive Binary Arithmetic Coding (CABAC) [7].

CABAC is a method of arithmetic coding in which the probability models are updated based on the previous coding statistics. CABAC provides good compression performance through selecting probability models for each syntax element according to the element’s context, adapting probability estimates based on local statistics and using arithmetic coding.

Coding the data symbols involves the following steps [14]:

(1) Binarization - maps the syntax elements into binary symbols (0 or 1)

(2) Context modeling - estimates the probability of the bins

(3) Arithmetic coding - compresses bins to bits based on the estimated probability

2.1.3.6 In-loop filtering

HEVC uses two consecutive in-loop filters such as the de-blocking and the Sample Adaptive Offset (SAO) filters which are shown in Figure 2-1. They are used in encoding and decoding loops after inverse quantization and before saving the picture in decoded picture buffer.

De-blocking filter is applied to prediction block and transform block edges in order reduce the amount of visible block structures, which results from the block based nature of the coding scheme. This filter operates on the block edges with adaptive filter strength and adaptive filter length [8].

The Sample Adaptive Offset (SAO) is a sample based filtering operation which is operated on a CTU basis. This filter can be configured to be driven by sample values differences of a local neighborhood, or by the value range intensity value of the current sample. The filter operates on the samples in the slice and does not only consider block edges. SAO is applied for the output of de-blocking filter and it reduces the ringing artifacts.

The DCT in HEVC works well on flat areas, but fails on areas with noise, contours and other peculiarities of the signal. It is efficient for large size of blocks but it is less efficient for smaller sized blocks. Beginning from 16x16 transforms, visual artifacts are noticeable. The artifacts are more observable when the transform size increases. De-blocking can reduce artifacts on TB boundaries, while artifacts inside a TB can be reduced only by SAO. Hence, SAO can be applied when large transform sizes (32x32) are used [14].

2.1.4 Parallel tools

HEVC gives flexibility to use various in-built high level parallelization tools. This section describes the parallelization tools that can be used in HEVC such as slice, tile and wave-front parallel processing.

2.1.4.1 Slices

A slice is a partition of picture that can be decoded independently from other slices in the same picture. It can be the entire region of the picture or a portion of the picture. The minimum block structure that has to be present in a slice is a CTU. The main purpose of slice is to provide error resilience and parallel processing capability.

Slices breakdown the CTU dependencies such as in-picture prediction and entropy coding at the slice boundaries, resulting in reduced exploitation of spatial dependency in the picture and also reduced coding efficiency. The coding efficiency decreases as the number of slices in a picture increases. Also, the bitrate overhead increases due to multiple slice headers. In order to address these issues, HEVC introduces the concept of slice segments and slice segment subsets.

Each slice at CTU boundaries can be divided into one or more slice segments. The first slice segment is an independent slice segment, followed by all dependent slice segments (if any) which have reduced slice headers. Dependent slice segments within the slice do not break the in-picture dependencies across the CTU boundaries. The slice segment subsets in HEVC are obtained by fragmenting the coded slice data without the use of additional header data. Figure 2-10 represents segmentation of slices in HEVC.

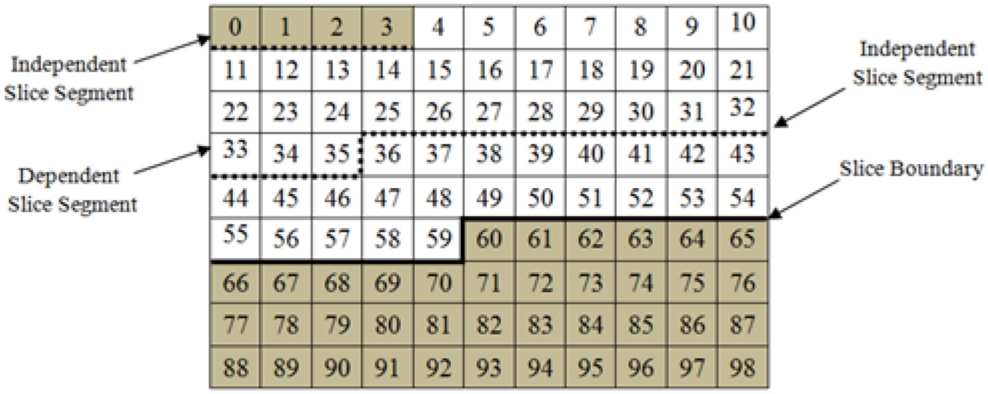


Figure 2-10 Segmentation of Slice in HEVC [13]

2.1.4.2 Tiles

Tile is a feature in HEVC that divides the picture into rectangular shaped groups of CTU's separated by vertical and/or horizontal boundaries as represented in Figure 2-10. Tiles break prediction and entropy coding dependencies across the tile boundaries, making them independent of each other. However, In-loop filtering might spread across different tile boundaries.

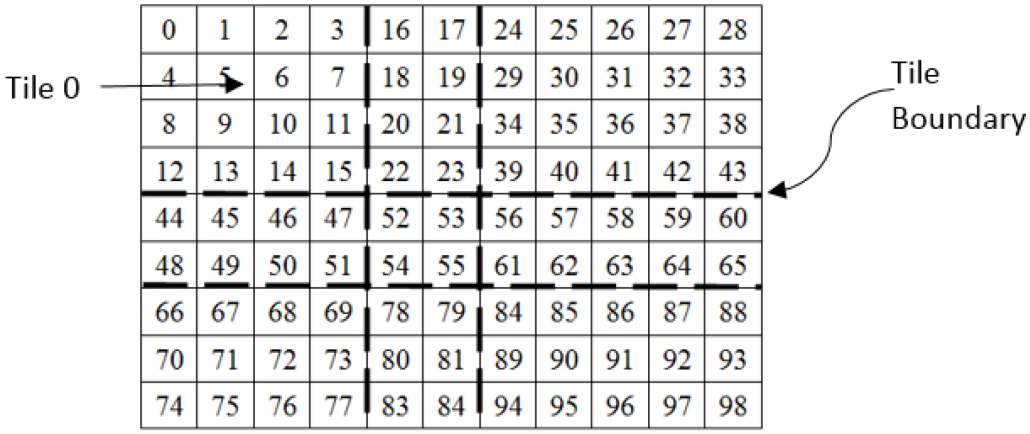


Figure 2-11 Tiles in HEVC [16]

As the number of tiles increases, the coding efficiency reduces due to breaking of dependencies across boundaries and also resets the CABAC context at the beginning of each tile. Tiles change the regular scan order of CTU's from frame-based raster scan order to a tile-based raster scan order. Figure 2-11 represents the tile structure in HEVC.

2.1.4.3 Wave-front Parallel Processing

Wave-Front Parallel Processing (WPP) is a parallelization technique, when used in HEVC splits the picture into rows of CTUs, where each CTU can be processed by different thread. The total number of threads that can be used when WPP is enabled depends on the number of CTU rows available, which in turn depends on ratio of picture height in luma samples and the luma CTB size. When WPP is used, each CTU row is processed relative to its previous row using a delay of two consecutive CTUs. This process is represented in the Figure 2-12.

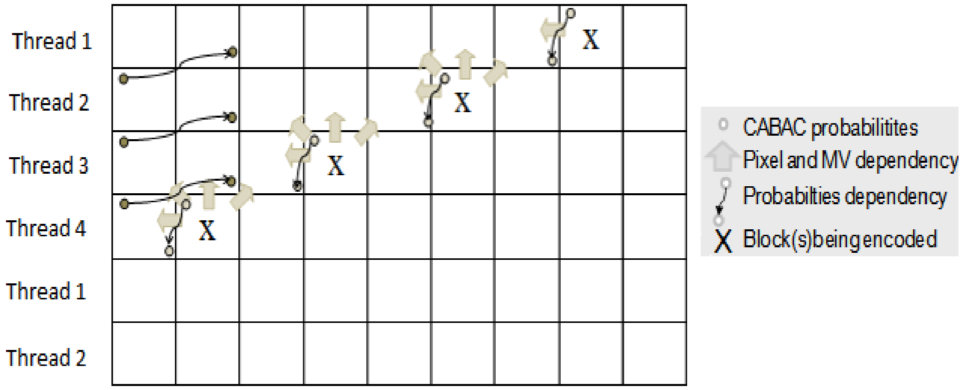


Figure 2-12 Wave-Front Parallel Processing in HEVC [17]

In WPP, no dependencies between consecutive CTU rows are broken at the partition boundaries except for the CABAC context variables at the end of each CTU row. The conventional CABAC re-initialization at the beginning of each row in this scenario would result in loss of coding efficiency. However, this is reduced by propagating the adapted CABAC context variables from the encoded/decoded second CTU of previous row to the first CTU of the current row.

2.1.5 Rate Control in HEVC Reference Software

Rate control involves modifying the encoding parameters in order to maintain a target output bitrate. The parameter to vary for this purpose is the quantizer parameter or step size (QP), since increasing QP reduces coded bitrate (at the expense of lower decoded quality) and vice versa. A common approach to rate control is to modify QP during encoding in order to (a) maintain a target bitrate (or mean bitrate) and (b) minimize distortion in the decoded sequence. Optimizing the tradeoff between bitrate and quality is a challenging task in rate control algorithm. The choice of rate control algorithm depends on the nature of the video application [10].

To measure the distortion between reconstructed and original picture on a sample to sample basis, objective measures are used. Various distortion measures used in HEVC reference software include Sum of squared differences (SSD), Sum of absolute differences (SAD), Sum of Absolute Transformed differences (SATD) and Peak Signal to Noise Ratio (PSNR) [6].

2.1.6 High Level Syntax

An HEVC bit-stream consists of a sequence of data units called network abstraction layer (NAL) units. There are two classes of NAL units in HEVC - video coding layer (VCL) NAL units and non-VCL NAL units. Each VCL NAL unit carries one slice segment of coded picture data while the non-VCL NAL units contain control information that typically relates to multiple coded pictures. There are 64 different NAL unit types. One coded picture, together with the non-VCL NAL units that are associated with the coded picture, is called an HEVC access unit.

The syntax elements that describe the structure of the bit-stream or provide information that applies to multiple pictures or to multiple coded block regions within a picture, such as the parameter sets, reference picture management syntax, and

Supplemental Enhancement Information (SEI) messages, are known as the “high level syntax” part of HEVC [6] [8]. Figure 2-13 represents the high level syntax of HEVC.

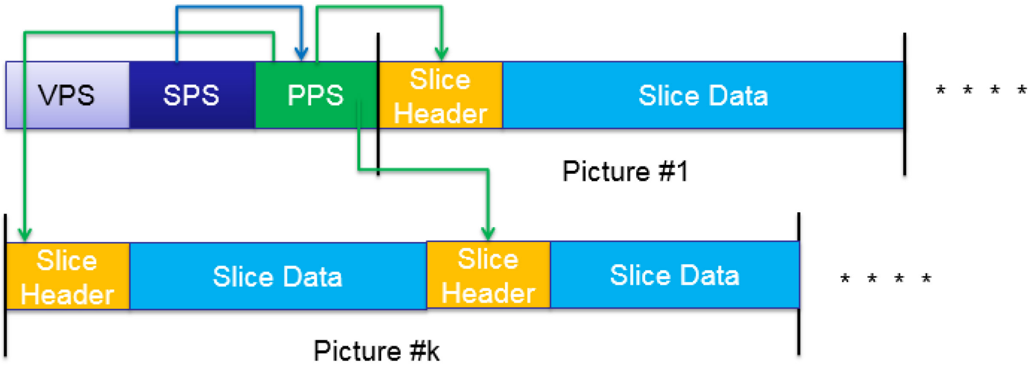


Figure 2-13 High Level Syntax of HEVC [14]

HEVC specifies three parameter sets. These are the video, sequence, and picture parameter set (VPS, SPS, and PPS). VPS collectively provides information on different layers and sub-layers of the coded video sequence. SPS contains information which applies to all slices of a video sequence and is fixed within this sequence such as profile, level, picture size, number sub-layers, enabling flags, restrictions, temporal scalability control and visual usability information (VUI). PPS conveys information which could change from picture to picture such as reference list size, initial QP, enabling flags and tiles or wave-fronts. Slice header conveys information that can change from slice to slice such as Picture Order Count (POC), slice type, prediction weights, de-blocking parameters and tiles entry points [14].

2.1.7 Profiles, Levels and Tiers

A profile defines the set of coding tools which can be used to encode a video sequence into a bit-stream. An encoder for a profile may choose which coding tools to use as long as it generates a conforming bit-stream while a decoder for a profile must support all coding tools that can be used in that profile. Version 1 of the HEVC standard defines three profiles: Main, Main 10, and Main Still Picture. Version 2 of HEVC adds 21 range extensions profiles, two scalable extensions profiles (Scalable Main and Scalable Main 10), and one multi-view profile [25].

The levels indicate restrictions on parameters which determine decoding and buffering capabilities. These include the maximum picture size, the coded and decoded picture buffer sizes, the number of slice segments and tiles in a picture, as well as the maximum sample rate and maximum bitrate. It further sets a requirement for the minimum compression ratio which must be met by a bit-stream. There are 13 levels defined in HEVC standard.

The concept of tiers enables the differentiation between different application types which require different available bitrate ranges. Correspondingly, the maximum bitrate and the maximum CPB size differ between tiers. Two tiers of levels are specified in HEVC. The Main tier targets consumer applications, the High tier is designed for professional applications [63].

2.2 Dynamic Adaptive Streaming over HTTP

2.2.1 Adaptive Bitrate Streaming

Adaptive bitrate streaming is a technique used for multimedia streaming over the computer networks. In the past, the streaming technologies used streaming protocols such as Real Time Protocol (RTP) and Real Time Streaming Protocol (RTSP) for streaming audio and video content. However, for efficient delivery of multimedia content in large segments, adaptive streaming technologies based on HTTP are used today and they are designed to work efficiently over large distributed HTTP networks such as the Internet [18].

Adaptive bitrate streaming works by detecting a user's bandwidth and CPU capacity in real time and adjusting the quality of a video stream accordingly. It uses an encoder that encodes a single video at multiple bit rates. Each of the different bit rate streams are segmented into small multi-second parts. The client switches between streaming various bit rate versions depending on the available resources [64].

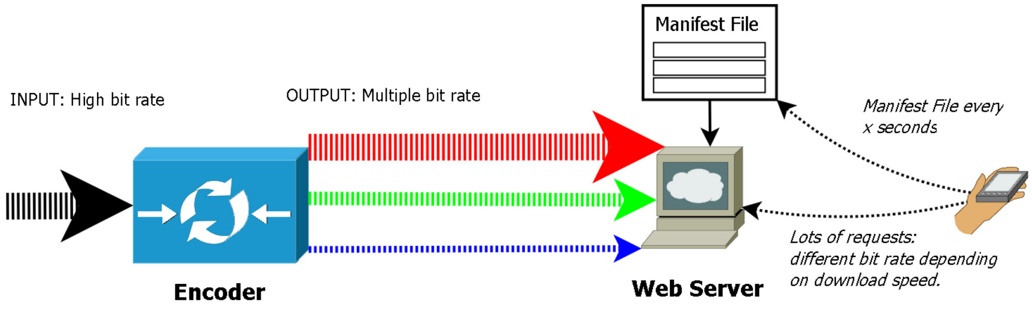


Figure 2-14 Overview of Adaptive Bitrate Streaming [64]

The three popular streaming technologies – Apple's HTTP Live Streaming (HLS), Adobe's HTTP Dynamic Streaming (HDS) and Microsoft's Smooth Streaming, use HTTP streaming as their delivery method and they are incompatible with other. In this scenario, video distributors had to support all the technologies and it is time consuming, complex and cost inhibitive. Hence, a standard was developed by MPEG – Dynamic Adaptive Streaming over HTTP (MPEG-DASH), which aims at streaming video delivery to various devices or platforms by ensuring interoperability between devices and servers from different vendors.

2.2.2 Overview of MPEG-DASH

Dynamic Adaptive HTTP Streaming (DASH), also known as MPEG-DASH [18] is an international standard developed by MPEG. DASH is audio or video codec agnostic. It uses an adaptive bit-rate streaming technique that enables high quality streaming of media content over the Internet delivered from the conventional HTTP web servers.

MPEG-DASH works by breaking the multimedia content into a sequence of small HTTP-based file segments, each segment containing a short interval of playback time of content that is potentially many hours in duration, such as a movie or the live broadcast of a sports event. The content exists on the server in two parts: Media Presentation Description (MPD), which describes a manifest of the available content, its various alternatives, their URL addresses, and other characteristics; and segments, which contain the actual multimedia bit streams in the form of chunks, in single or multiple files. To play the content, the DASH client first obtains the MPD [18].

When the current content is being played back by a DASH client, the client automatically selects from the alternatives the next segment to download. By parsing the MPD, the DASH client gets information about the program timing, media-content availability, media types, resolutions, minimum and maximum bandwidths, and the existence of various encoded alternatives of multimedia components, media-component locations on the network, and other content characteristics. Using this information, the DASH client selects the appropriate encoded alternative and starts streaming the content by fetching the segments [18]. A sample streaming scenario between HTTP server and DASH client is represented in Figure 2-15.

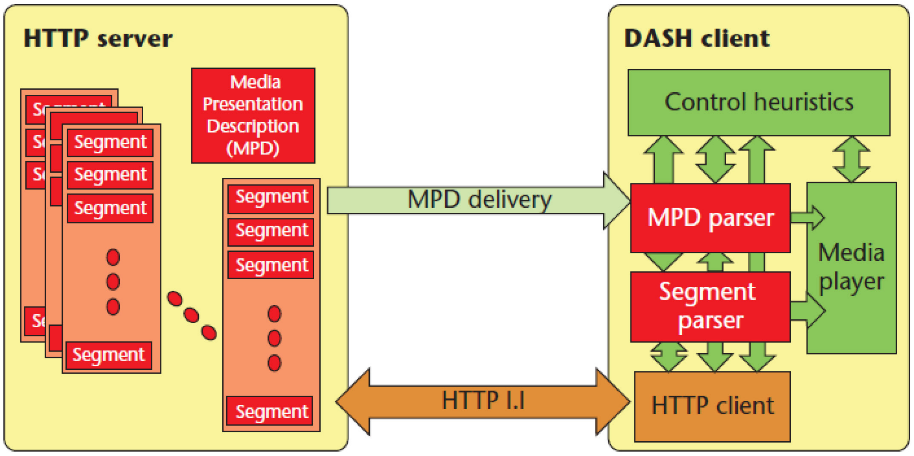


Figure 2-15 Streaming Scenario between HTTP Server and DASH Client [18]

2.3 Ultra-High Definition Video

High Definition (HD) [30] has become available almost everywhere in the world. Additionally, in the last couple of years, new Ultra High Definition (UHD) [28] [29] video format is gaining commercial importance to provide next generation video services. UHD format defines enhanced parameters in multiple aspects of a video signal.

UHD supports higher spatial resolutions of 3840x2160 (4K or UHD-1) and 7680x4320 (8K or UHD-2) image samples and higher frame rates up to 120 Hz than HD; also bit depths up to 12 bits for high dynamic range support and a wider color gamut that enables rendering of more vivid colors. The term UHD or 4K primarily refers to UHD-1 format. Table 2-1 provides a comparison of HD and UHD video parameters [27].

Figure 2-16 shows the HD and UHD color gamut overlaid with the International Commission on Illumination 1931 color space chromaticity diagram. The horseshoe shape represents the range of colors visible to human eyes. The narrower triangle is the BT.709 [30] color gamut and the wider triangle is the BT.2020 [28] color gamut. The potential benefits of UHD exists in the wider color gamut and higher bit depth. The 8 bit BT.709 (HD) signal is capable of representing 16.8 million colors. In comparison, 10-bit and 12-bit BT.2020 (UHD) signals are capable of representing 1.07 billion and 68.7 billion colors, respectively.

Table 2-1 Comparison of HD and UHD video formats [27]

		High Definition	**Ultra-high Definition**
ITU-T BT series		BT.709-5(part 2) [30]	BT.2020 [28]
Spatial		1920x1080	7680x4320 3840x2160
Temporal	Frame rate (fps)	60,50,30,25,24	120, 60, 50, 30, 25, 24
	Scan	Progressive, Interlaced	Progressive
Primary colors	Red primary	(0.640, 0.300)	(0.708, 0.292)
	Green Primary	(0.150, 0.330)	(0.170, 0.797)
	Blue Primary	(0.600, 0.060)	(0.131, 0.046)
Coding format		8 and 10 bit	10 and 12 bit

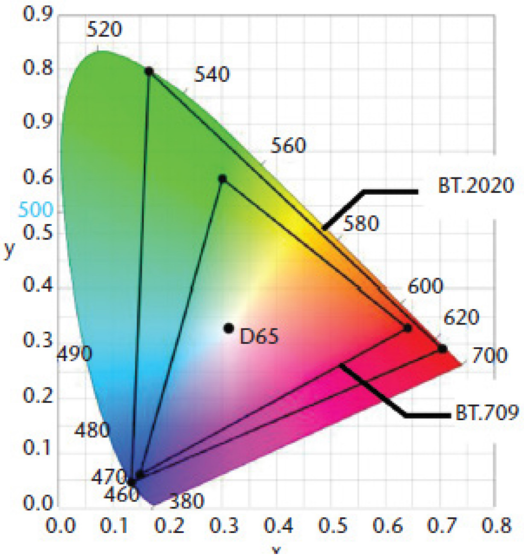


Figure 2-16 Comparison BT.709 and BT.2020 color gamuts [27]

For capturing and deploying UHD video, 4K cameras and 4K TV sets have become available in the market. Also, the content providers (broadcasters, studios, etc.) are exploring the production of native UHD content or remaster the existing content in the new UHD format. For UHD to be a success, the implementation of BT.2020 should not only support increased spatial resolution but also higher bit depth and wider color gamut [27].

2.4 Summary

This chapter gives an introduction of state of the art video coding and streaming technologies. Firstly, it presents an overview of the HEVC video coding standard and its coding principles such as the partitioning, prediction, transform, quantization, entropy coding and in-loop filtering. It also describes the parallel coding tools and rate control feature. The high level syntax of HEVC is also presented. Additionally, it introduces the adaptive bitrate streaming technology and briefly describes the MPEG-DASH standard. It also gives a brief description of UHD video format by comparing it to HD video format. The next chapter gives a brief introduction to scalable video coding and the SHVC standard.

Chapter 3

SCALABLE VIDEO CODING TECHNOLOGY

3.1 Scalable Video Coding

Scalable video coding is a technique where video is coded in multiple layers, and each layer represents a version of same video in terms of spatial resolution, temporal frame rate or the quality. Scalable stream has base layer (BL), which is the basic version of the video and enhancement layer(s) (EL) which are encoded on top of base layer containing enhanced versions of the video. There are three main scalability options available in scalable coding such as spatial, temporal and quality scalability.

There are different representations of video that useful for display types supporting different spatial resolutions, color representation and bit-depths; connection types having different link speeds (such as LTE, cable etc.) and constant and varying speeds; different processing powers in case of mobile and stationary devices. Traditional approach used in this scenario is to encode and store video streams supporting possible clients and connection speeds [25]. The benefits of scalable video coding are applicable in these scenarios where different versions of the same video content are necessary.

Scalable coding finds its application in video broadcasting, where it can provide compatibility for old devices and at the same time improve experience for new devices; video streaming and conferencing applications, where it can provide efficient adaptation to different devices and changing network conditions [25]. It is also highly desirable for surveillance applications, in which video sources not only need to be viewed on multiple devices ranging from high-definition monitors to videophones or PDAs, but also need to be stored and archived [26].

The scalable coding has various advantages such as less storage space, less data rate in multi and broadcast environment, more flexibility because of combination of layers, built-in support of up and down switching and error resilience. However, these advantages of scalable coding come at a cost of additional overhead compared to single layer coding and increased complexity of decoder.

3.2 Overview of Scalable High Efficiency Video Coding (SHVC)

Scalable High Efficiency Video Coding (SHVC) [2] standard is the scalable extension of High Efficiency Video Coding (HEVC/H.265), jointly developed by ISO/IEC MPEG and ITU-T VCEG. The SHVC extensions were standardized in 2014. The previous scalable video coding standard was based on H.264/MPEG-4 AVC known as Scalable Video Coding (SVC) [26].

With SHVC, basic version of the video is coded as base layer and improved versions of the video is coded as enhancement layers. SHVC design philosophy enables SHVC implementation using multiple single-layered HEVC cores to achieve high scalable coding efficiency. SHVC system architecture requires high level syntax (HLS) changes in HEVC at slice header level and above [19]. SHVC reuses all the compression tools of HEVC along with interlayer prediction, which exploits the redundancies between layers to gain coding efficiency by predicting the enhancement layer using the base layer [21]. SHVC employs multi-loop coding framework where the BL is decoded first, followed by decoding the EL using interlayer prediction. SHVC supports resolution up to 4K [22] and 8K [23].

The following are different scalability options in available in SHVC [25]:

1. Temporal scalability – provides higher frame rates in EL

   Example: 30 fps in BL to 60 fps in EL

2. Spatial scalability – provides higher spatial resolution in EL

   Example: HD in BL to UHD in EL

3. Coarse grain SNR scalability – provides higher SNR qualities in EL

   Example: Low-quality BL at 1Mbps to high-quality EL at 8 Mbps

4. Bit depth scalability – provides higher bit depth in EL

   Example: BL with bit depth of 8 bits to EL with bit depth of 10 bits

5. Interlaced-to-progressive scalability – where BL can be in interlaced format and EL in progressive format

6. Color gamut scalability – provides higher color gamut in EL

   Example: BT.709 [30] color gamut in BL to BT.2020 [28] color gamut in EL

7. External base layer scalability – where base layer is encoded by another external encoder

   Example: MPEG-4 AVC/H.264 BL with SHVC EL

8. Combination of these scalabilities

The HEVC design provides temporal scalability when hierarchical temporal prediction is used. The other scalability features in SHVC are enabled using the layered approach [19].

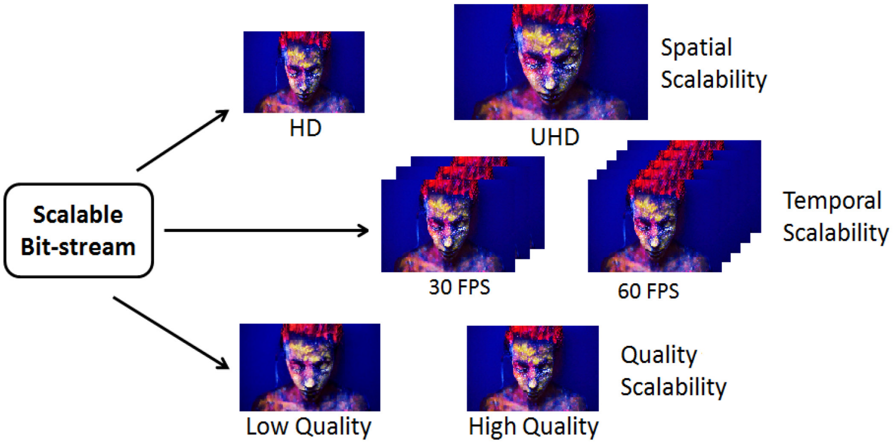


Figure 3-1 Few Scalability options in SHVC [3]

In today’s multimedia environment, UHD video content is gaining importance, but it will not completely replace HD contents because of increased data rates and backward compatibility with legacy devices [20]. In particular, backward compatibility with legacy devices can be supported using SHVC. UHD contents can be encoded as the enhancement layer (EL) of HD contents, so legacy devices capable of decoding HD contents can be used continuously, while new devices can decode both UHD and HD contents. This is supported by hybrid codec feature of SHVC, where the base layer can be encoded with HEVC or a non-HEVC codec such as MPEG-4 AVC.

3.3 SHVC Architecture

SHVC encodes the original input video into L layers. The first layer represents the base quality of video, and decoding more layers allows to further enhance the video. It uses a multi-layered decoding structure. To decode L^{th} layer, all intermediate layers from l = 1 to L-1 have to be fully decoded to perform inter-layer predictions. Figure 3-2 represents the architecture of SHVC for HTTP streaming. If base layer is encoded with HEVC, then HEVC decoder is used. Similarly, if base layer is encoded with non-HEVC encoder such as MPEG-4 AVC, then MPEG-4 AVC decoder is used.

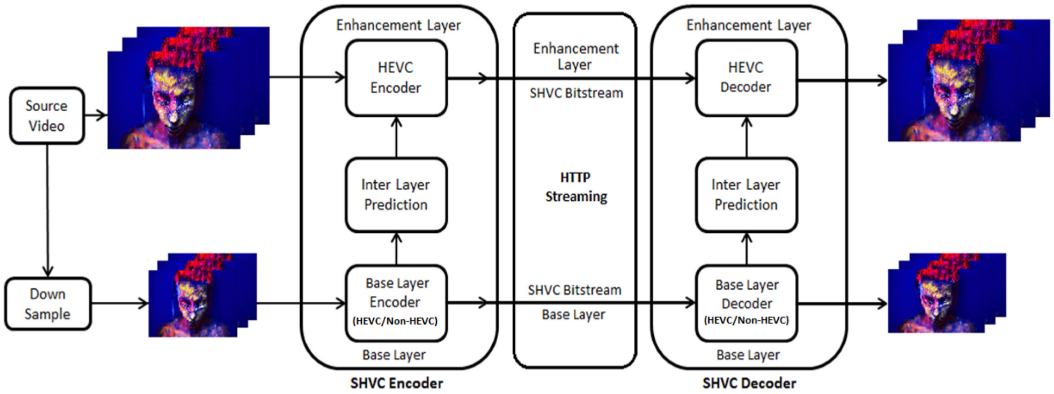


Figure 3-2 SHVC Encoder and Decoder Architecture [21]

For spatial coding with L-layers, the SHVC encoder consists of ‘L’ HEVC encoders, one for each layer. The Base Layer (BL) HEVC or non-HEVC (H.264/MPEG-4 AVC) encoder (l=1) encodes the down-sampled version of the original video and feeds the HEVC encoder corresponding to next Enhancement Layer (EL, l=2) with the decoded picture and its motion vectors (MVs). The L^{th} HEVC encoder encodes the original video using the up-sampled picture from the lower layer HEVC encoder (l=L-1) and its up-scaled MVs as an additional reference picture for interlayer predictions.

For the scenario of UHD deployment with two layers in SHVC supporting spatial scalability, the original video of UHD resolution is down sampled to HD resolution and encoded as the base layer. A spatial enhancement layer is further added to encode the UHD resolution video.

The decoding in SHVC is designed to be multi-loop and for inter layer prediction, all the samples from the reference layers within the specified reference are used. As SHVC uses multi-loop design, the SNR or quality scalability in SHVC when compared to single-layer HEVC has higher complexity cost, as two or more layers are coded at the same resolution. However, for spatial scalability, the additional complexity cost of SHVC over single layer HEVC is lower, as lower layers have lower resolution [4].

3.4 Multi-layer High Level Syntax

The common multilayer high level syntax design for scalable and Multiview extensions such as SHVC and MV-HEVC is given in HEVC version 2 [25]. The SHVC bit-stream consists of layers having BL and ELs. The BL can be based on HEVC or non-HEVC.

HEVC has a new concept in multi-layer bit-stream called the layer set, which is defined as a set of layers that forms the decodable sub-bitstreams. A layer set can be defined with two layers: Layer 0 (BL) and Layer 1 (EL). Another layer set can be defined to have only Layer 0 (BL). However, a layer set cannot have only layer 1 (EL), as EL needs BL for its decoding. An output layer set is defined as a layer set with an associated set of target output layers. The target output layers of a layer set specify which layer(s) a decoder will output. For the layer set in Figure 3-3, Layer 1 (EL) could be set as the only target output layer [4]. Figure 3-3 gives sample representation of a bit-stream with layers.

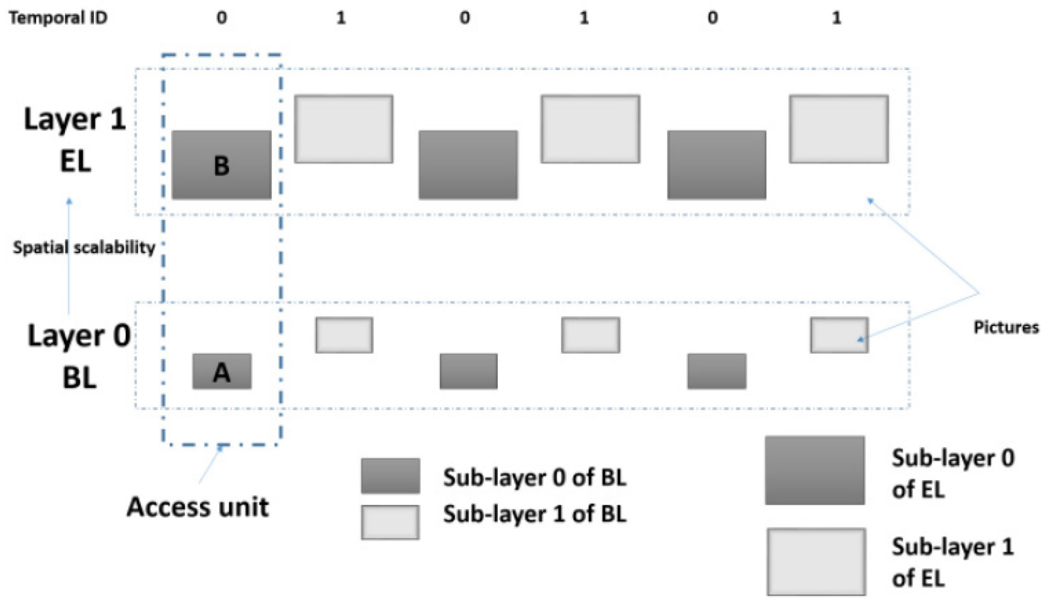


Figure 3-3 Sample Layering and Sub-Layering in SHVC Bit-stream [4]

The Picture Order Count (POC) is the relative order of the output pictures used for indicating the reference picture. SHVC's temporal scalability is referred to temporal sub-layers. Operation point defines sub-sets of temporal sub-layers that can be decoded. Temporal ID value identifies temporal prediction restrictions among the pictures, where in a particular temporal ID value cannot use a picture with higher temporal ID for reference. An access unit contains one or more coded pictures, each from a different layer, which are associated with the same instant of output time [4].

The SHVC layer related information is indicated in NAL unit header and Video Parameter Set (VPS). The NAL unit header contains information about the layer and temporal sub-layer associated with the NAL unit. The VPS contains information about the layer types, layer dependencies, layer sets, operation points, output layer sets and layer representation or hypothetical reference decoder or decoded picture buffer information.

3.4 Coding of Enhancement Layer in SHVC

The enhancement layer in SHVC is predicted from a base layer using a powerful tool called inter-layer prediction, which exploits redundancies between layers to improve coding efficiency. The high-level syntax design of SHVC is modified such that the collocated reconstructed pictures (resampled if necessary) from the reference layers (i.e., reference layer pictures with the same POC value as that of the current picture) can be used as inter-layer reference pictures when coding the current EL picture. This allows inter-layer prediction to be carried out without any low level coding process changes.

Interlayer processing in SHVC is applied to the reference layer (RL) pictures to form Interlayer reference (ILR) pictures when any parameters such as the spatial resolution, bit depth and color gamut between the RL and the current EL are different [4]. Figure 3-4 represents interlayer prediction used in SHVC for spatial scalability, where BL picture is up-sampled and used as interlayer reference (ILR) picture along with temporal references for predicting EL.

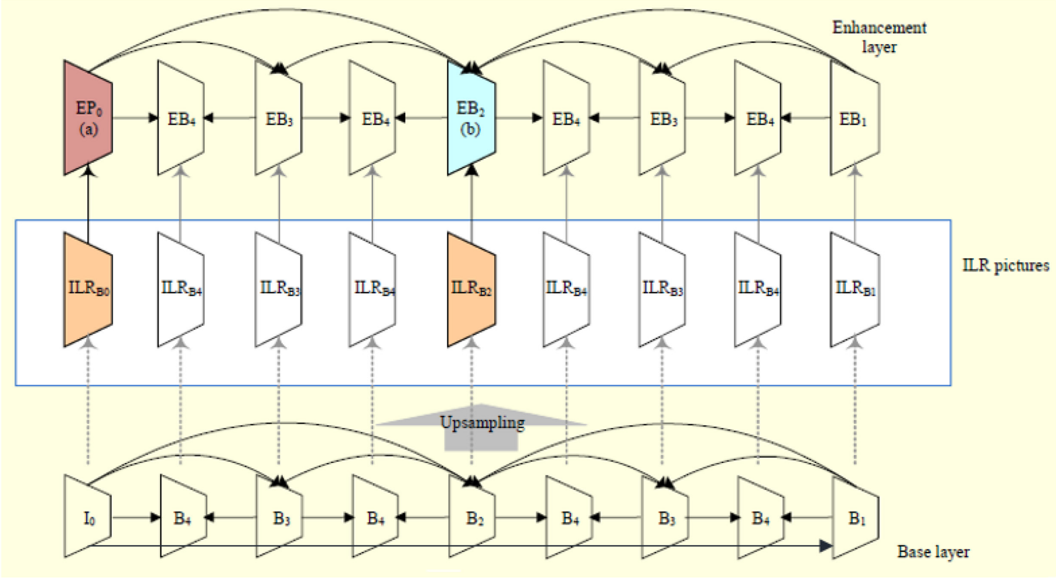


Figure 3-4 Inter-layer Prediction in SHVC [24]

The EL in SHVC can apply multiple ILRs for interlayer prediction. However to reduce the complexity of inter layer processing during decoding, only one ILR picture may require resampling when EL is decoded. There is no need for storing ILR pictures once the current EL picture is decoded, as interlayer references occur within current access unit [4].

In order to generate ILR pictures, three main modules are used - texture resampling and motion field resampling for spatial and bit depth scalability; color mapping for color gamut scalability [20]. These are the three main blocks which are added to the HEVC codec in order to support SHVC codec and these changes occur above slice header level and above [19].

3.4.1 Interlayer Texture Prediction in SHVC

In SHVC, inter-layer texture prediction is invoked by including the ILR pictures from the reference layers (with resampling and color mapping process performed if necessary), together with the temporal reference pictures, in the reference picture lists of the EL picture. At the Prediction Unit (PU) level, the signaled one or two reference picture indices are used to indicate whether the current PU is predicted from temporal reference pictures, from ILR pictures, or from a combination of both. When a PU is predicted from at least one ILR, there is a bit-stream conformance constraint that requires the motion vectors associated with the ILR picture(s) to be zero.

The initial reference picture lists in SHVC are constructed as follows: For reference picture list 0 (L0), the ILR picture(s) is inserted between the set of short-term temporal reference pictures for forward temporal reference pictures and the set of short-term temporal reference pictures for backward temporal reference pictures. For reference picture list 1 (L1), the temporal references are first added into the reference list in the same manner as the initial reference picture list construction in HEVC. After that, the ILR picture(s) is added at the end of L1 as long term reference picture(s). The ILR picture(s) is added to the reference picture list L0 when the current EL picture is coded as P-Slice, and is added to both reference picture lists L0 and L1 when current EL picture is coded as B-Slice [19].

3.4.2 Interlayer Motion Prediction in SHVC

In SHVC, motion field mapping is the process of using the reference layer motion information when coding the enhancement layer motion vectors by making use of the existing Temporal Motion Vector Prediction (TMVP) process of HEVC.

In HEVC, TMVP is used to predict motion information for a current PU from a co-located PU in the reference picture. The process requires prediction modes, reference indices, luma motion vectors and reference picture order counts (POCs) of the co-located PU. This information is stored on a 16x16 luma block basis, which may be a lower resolution than what is transmitted in the bit-stream in cases of small PU sizes. The motion field mapping process projects this motion information from the reference layer to the enhancement layer's resolution, while also accounting for 16x16 TMVP storage units in the reference layer [3].

The first step in the mapping of the motion information is to determine for the current enhancement layer PU the co-located position in the stored reference layer motion information, taking into account the reduced motion information storage resolution as well as the up-sampling ratio between the two layers and any reference layer offsets. Once the co-located position is determined and the motion information from the co-located reference layer PU is available, a scaling operation is applied to those motion vectors to account for the up-sampling ratio, since motion vectors also grow with the picture resolution. However, no further scaling depending on temporal distance is applied due to the fact that the reference layer picture is indicated as long term picture [3].

As shown in Figure 3-5, where each grid in the enhancement layer (right) picture represents an 8×8 block and each grid in the reference (left) layer represent a 4×4 block, the collocated 16×16 block in the reference layer picture is derived as follows [19]:

1. The collocated sample location of the center sample of the 16x16 block in the reference layer picture is denoted as (x_{RL}, y_{RL}).
2. The location (x_{RL}, y_{RL}) is then rounded to align with a 16x16 block by using an offset of 4, as follows,

$$x_{RL} = ((x_{RL} + 4) >> 4) << 4 \quad (3.1)$$

$$y_{RL} = ((y_{RL} + 4) >> 4) << 4 \quad (3.2)$$

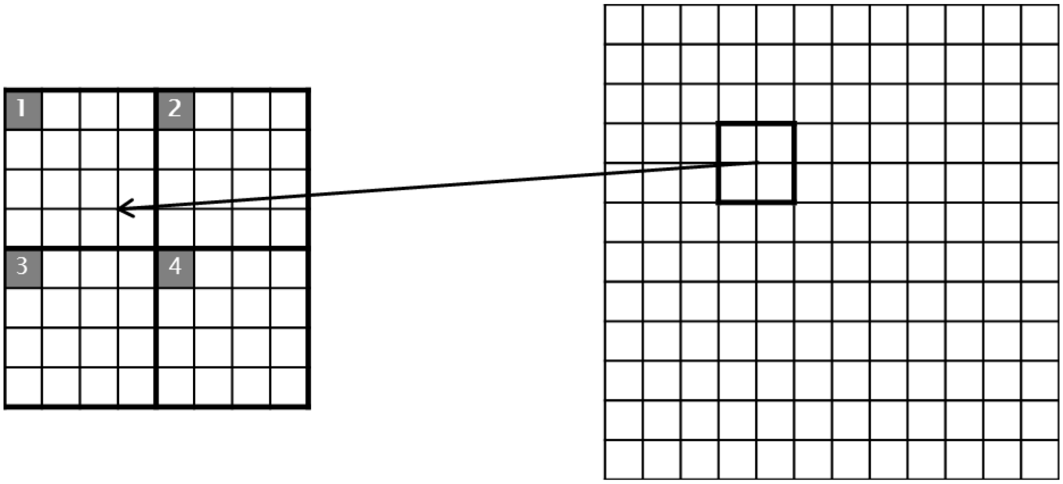


Figure 3-5 Derivation of collocated 16×16 block in reference layer

As shown in Figure 3-5, the values of (x_{RL}, y_{RL}) are rounded to align with a 16x16 block with top-left location indicated by 1, 2, 3 or 4. When the sample position (x_{RL}, y_{RL}) is located outside the reference layer picture, the motion information of the current 16x16 block is marked as unavailable by setting block prediction mode to intra prediction mode.

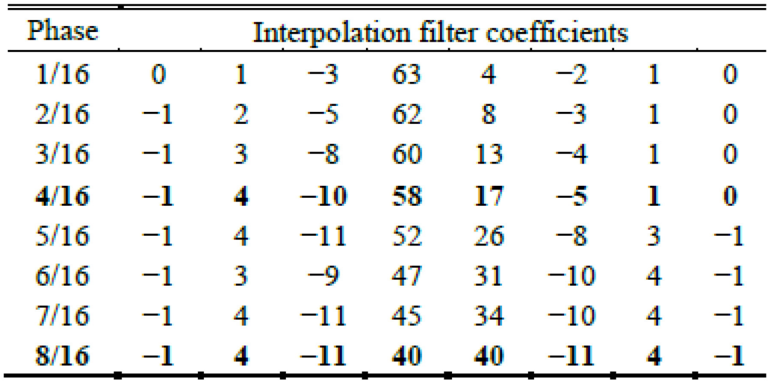
The motion mapping process can be enabled or disabled within the bit-stream, and it is disabled when base layer is provided from external means [3].

3.4.3 Interlayer Reference Picture Processing Tools

3.4.3.1 Up-sampling filters

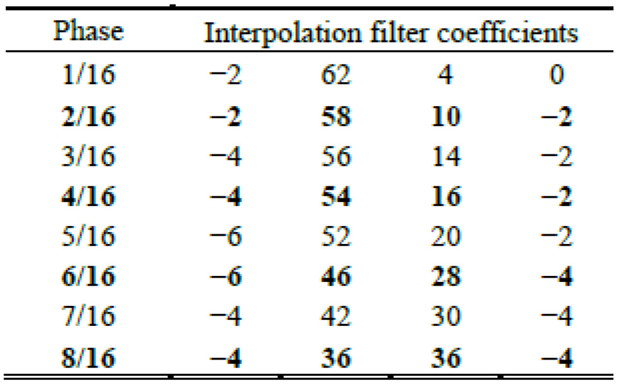
The up-sampling filter in SHVC is used to map reconstructed sample values from the reference layer to the higher-resolution sampling grid of the enhancement layer. This allows the use of the reconstructed RL sample values for EL prediction. The up-sampling process is defined as a normative part of the standard. The down-sampling process is used to create the source pictures of lower resolution as input to the encoding process of the reference layer is outside the scope of the standard [3].

The up-sampling filters are a set of filters with 1/16th fractional sample accuracy. SHVC uses 8-tap poly-phase finite-impulse-response (FIR) filter for luma resampling and a 4-tap poly-phase FIR filter for chroma resampling. The up-sampling filters are designed such that they align with fractional sample motion interpolation filter which has 1/4th sample accuracy for luma and 1/8th sample accuracy for chroma in first version of HEVC. Figures 3-6 and 3-7 represents half of phase filters as other half have filter coefficients which are symmetrical to the ones shown [4].



Phase	Interpolation filter coefficients							
1/16	0	1	−3	63	4	−2	1	0
2/16	−1	2	−5	62	8	−3	1	0
3/16	−1	3	−8	60	13	−4	1	0
4/16	**−1**	**4**	**−10**	**58**	**17**	**−5**	**1**	**0**
5/16	−1	4	−11	52	26	−8	3	−1
6/16	−1	3	−9	47	31	−10	4	−1
7/16	−1	4	−11	45	34	−10	4	−1
8/16	**−1**	**4**	**−11**	**40**	**40**	**−11**	**4**	**−1**

Figure 3-6 Luma up-sampling interpolation filters in SHVC [4]



Phase	Interpolation filter coefficients			
1/16	−2	62	4	0
2/16	**−2**	**58**	**10**	**−2**
3/16	−4	56	14	−2
4/16	**−4**	**54**	**16**	**−2**
5/16	−6	52	20	−2
6/16	**−6**	**46**	**28**	**−4**
7/16	−4	42	30	−4
8/16	**−4**	**36**	**36**	**−4**

Figure 3-7 Chroma up-sampling interpolation filters in SHVC [4]

Scaled reference layer offsets can be signaled to enable the reference layer and enhancement layer the freedom to not fully correspond to the same region of a picture. Scale factors for the horizontal and vertical directions are computed as the ratio between the relevant enhancement and reference layer regions widths and heights, respectively. For each enhancement layer sample, the corresponding reference layer sample location and 1/16 sample phase is determined considering the scale factors and the scaled reference layer offsets. The 8-tap (or 4-tap) filter coefficients which correspond to the calculated phase are applied to the input reference layer samples, which are the sample at the reference sample location and its neighboring samples in the reference layer [3].

3.4.3.2 Color mapping

Color gamut scalability refers to the scalability use case when the RL and the EL have different color gamuts, typically with the EL having wider color gamut than the RL. In this case, SHVC applies a color mapping process to improve the coding efficiency. A 3D look-up table (LUT) based color mapping is used to generate texture samples in the ILR picture by converting samples in the RL picture from the RL color space to the EL color space. When spatial scalability and color gamut scalability are used in combination, both up-sampling and color mapping are required to generate the ILR picture. In this case, color mapping is applied prior to up-sampling to reduce computational complexity by performing color mapping on the lower-resolution pictures [4].

3.5 Summary

This chapter gives an introduction to scalable video coding and its applications. It discusses about the scalable extension of HEVC – SHVC, the different scalability options and architecture of SHVC. The multilayer syntax design for SHVC is discussed. Further, the interlayer prediction scheme is briefly discussed, along with inter-layer texture and motion prediction schemes. The up-sampling filters and the color mapping process in SHVC are also discussed. The next chapter describes the evaluation of SHVC for UHD deployment in the context of HTTP streaming.

Chapter 4

SHVC EVALUATION FOR UHD DEPLOYMENT IN THE CONTEXT OF HTTP STREAMING

4.1 Need for Evaluation

Typically, multiple versions of the same video are stored for streaming to clients having varying device capabilities such as displays, processing power and codec support, and also varying network characteristics. Using scalable coding in this scenario results in a single bit-stream having multiple layers i.e. one base layer and one or more enhancement layers, which can be used to cater to this heterogeneous client distribution. Hence, in this scenario, spatial and quality scalability of SHVC are evaluated for deploying UHD.

Although the UHD video and its services are increasing in the multimedia ecosystem, the older devices are still supporting HD video. Some clients support early codecs such as MPEG-4 AVC and the other clients support newer codecs such as HEVC. Thus, to provide backward compatibility and to deliver UHD to a mix of this HD and UHD clients, the hybrid codec feature of SHVC, where the base layer can be MPEG-4 AVC or HEVC HD layer and enhancement layer can UHD SHVC layer is evaluated.

When scalable video coding is used, the information present in the base layer is not duplicated in the enhancement layer. This obviates the need for storing multiple versions of the same video. The resulting savings are referred to as “bitrate savings”. However, this bitrate savings comes at a cost of reduced coding efficiency due to separation of BL and EL and additional protocol headers, which is called the “scalability overhead” [21].

The bitrate savings and scalability overhead for UHD deployment in the context of HTTP streaming are mainly investigated. The results are applicable for other contexts as well. Here, SHVC with 2 layers is evaluated for spatial scalability when there is a mix of HD and UHD clients. Also, SHVC with 3 layers of scalability, having HD and UHD spatial layers and an additional UHD quality scalability is evaluated for clients with varying bandwidth characteristics. These evaluations are performed for the case of MPEG-4 AVC and HEVC as base layers.

4.2 Evaluation Metrics

4.2.1 Rate Distortion Plots

When evaluating coding performance of a video codec, a graph of Rate–Distortion (R–D curve) is used. R–D curve is generated by plotting the encoded results, in terms of bit rate versus the resulting quality, in a graph. The horizontal axis denotes the bit rate and the vertical axis denotes a measure of distortion or quality of encoded video. In general, a higher compression ratio results in a lower bit rate; however, picture quality is generally reduced. Low compression ratio, on the other hand, improves picture quality but at the cost of an increase in bit rate. Since a high coding efficiency codec can achieve higher quality at lower bit rates, the R–D curve moves toward upper left, as shown in Figure 4-1 [6].

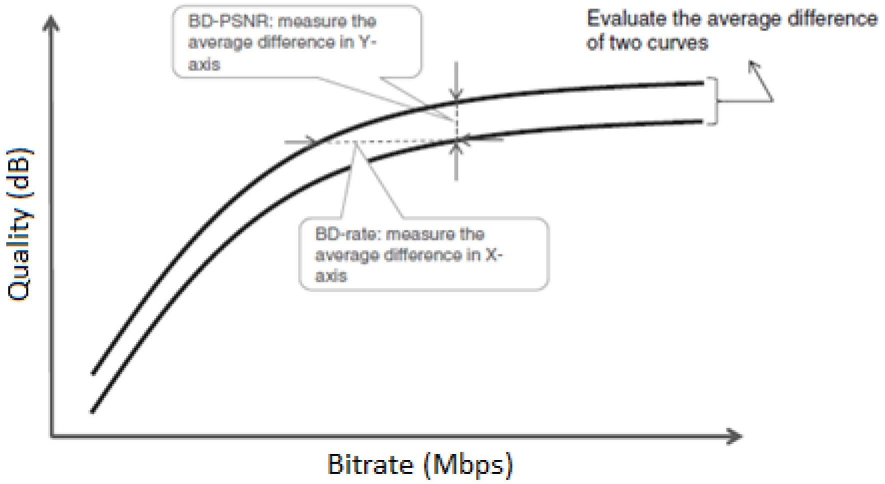


Figure 4-1 Sample R-D Curve [6]

Peak Signal to Noise Ratio (PSNR) is used as an objective measurement of picture quality. It can be calculated by the following equation:

$$PSNR = 10\log_{10}\frac{\left(2^{bitdepth}-1\right)^2 * W * H}{\sum_i \{O_i - D_i\}^2} \qquad (4.1)$$

where:

bitdepth - Bit depth of each pixel

W - Number of horizontal pixels

H - Number of vertical pixels

O_i - Pixel value of the reference picture

D_i - Pixel value of the decoded picture

i - Pixel address

PSNR is calculated for each YC_bC_r component. For 4:2:0 chroma sub-sampling format, total PSNR is computed as a weighted average of PSNRs of luminance ($PSNR_Y$) and chrominance ($PSNR_U$, $PSNR_V$) components as given below:

$$PSNR = \frac{6 \times PSNR_Y + PSNR_U + PSNR_V}{8} \qquad (4.2)$$

4.2.2 BD-Metrics

In order to compare the coding efficiency of a reference codec and the one being evaluated, the average difference of the two R–D curves is calculated using Bjontegaard's Delta metrics [52]. The average bit rate difference horizontal direction between the R-D curves is referred to as BD-Bitrate and the average PSNR difference in vertical direction between R-D curves is referred to as BD-PSNR [52].

In order to calculate BD-Rate and BD-PSNR, the two R–D curves corresponding to reference and tested codecs are approximated by the following cubic polynomial.

$$\text{PSNR} = a + b*(\text{bitrate}) + c*(\text{bitrate})^2 + d*(\text{bitrate})^3 \qquad (4.3)$$

Parameters a-d in Equation 4.3 can be derived by using four data points of PSNR and bit rate points. The polynomial approximation then allows to derive the BD-Rate by integrating the difference of two curves in horizontal direction and BD PSNR by integrating the difference of two curves in vertical direction as seen in Figure 4-1 [6].

BD-Rate and BD-PSNR metrics are widely used to evaluate coding tools in the video codec. The computation of BD metrics with more than 4 R-D points is performed using [55].

4.3 Evaluation Methodology

The R-D plots for all the methods of coding are obtained by incrementally varying the encoding bitrates. For example – the total bitrate for SHVC encoding is varied from 8 Mbps to 40 Mbps and resulting PSNR in dB is plotted against the bitrates. The bitrate savings obtained by using SHVC coding is obtained by comparing with simulcast coding. In simulcast coding, different versions of the video are coded independently and stored. The scalability overhead is obtained by comparing SHVC coding with HEVC single layer coding. The experiments are repeated for SHVC with 2 layers of spatial scalability and SHVC with 3 layers of combined scalability.

The 2 layered spatial scalability is investigated by encoding HD video as base layer (using HEVC or MPEG-4 AVC) and UHD video as enhancement layer of SHVC. For 3 layers, the combined spatial and quality scalability is investigated using HD as BL, UHD as EL1 and quality enhanced UHD as EL2.

An example comparison of encoding bitrates of SHVC with simulcast and single layer coding is given in Table 4-1. Here, B_{BL} and B_{EL} are the bitrates of the base and enhancement layers of SHVC. B_{LR} and B_{HR} are the bitrates of the lower and higher version of the video in simulcast or single layer coding. In all the coding methods, common base layer is used. If SHVC layers has bitrates of B_{BL} = 6Mbps and B_{EL} = 6 Mbps, two video are decoded at 6 Mbps and 12 (6+6) Mbps. As seen from Table 4-1, SHVC results in bit rate savings of 6 Mbps when compared to simulcast coding. However, this comparison is not fair as the bit rate savings are obtained at different quality levels as plotted in Figure 4-2.

Table 4-1 Comparison of SHVC with simulcast and single layer coding

Coding Method	Bitrate of Base Version or Layer in Mbps	Bitrate of Enhanced Version or Layer in Mbps	Total Bitrate in Mbps
SHVC	$B_{BL} = 6$	$B_{EL} = 6$	12
Simulcast Coding	$B_{LR} = 6$	$B_{HR} = 12$	18
Single layer Coding	-	$B_{HR} = 12$	12

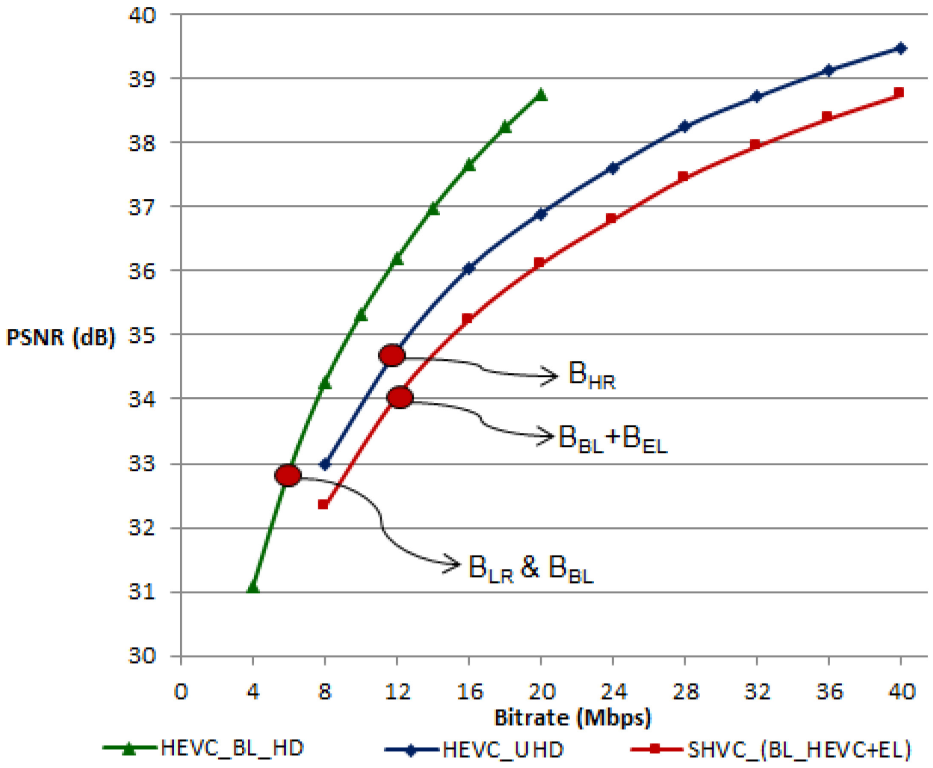


Figure 4-2 R-D Plot for comparison of SHVC with Simulcast and Single Layer Coding

In order to have a fair comparison, the bitrate differences between SHVC and simulcast coding are compared at same quality as represented in Figure 4-3. Using the previous example, the simulcast plot is obtained by adding the bitrate of HD (6 Mbps) and UHD (10.8 Mbps). The UHD curve is interpolated at the total bitrate of SHVC (BL+EL = 12 Mbps) and at similar quality level (34 dB) to obtain bitrate of 10.8 Mbps. This results in a bitrate saving of 6 (HD) + 10.8 (UHD) - 12 (SHVC) = 4.8 Mbps. Scalability overhead is the PSNR difference between the HEVC UHD plot and SHVC plot at the same bitrate of 12 Mbps, which is approximately 0.7 dB.

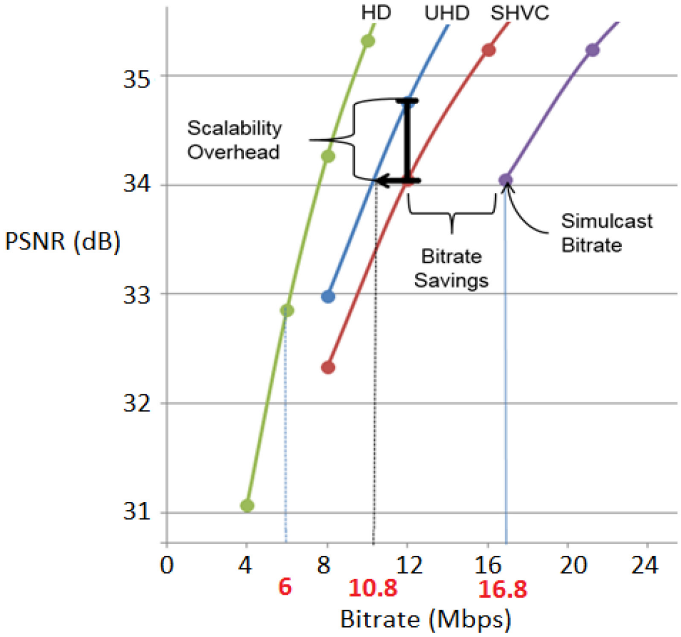


Figure 4-3 Bitrate comparison at Equal Quality

4.4 Experimental Set-up

4.4.1 Reference Software

In order to investigate SHVC, the SHVC reference software - SHM 6.1 is used for the encoding and decoding [57]. For single layer coding, HEVC reference software - HM 15.0 is used [58]. To encode the base layer with MPEG-4 AVC, the MPEG-4 AVC reference software - JM 18.6 is used [59].

4.4.2 Test Sequences

The standard YUV test sequences of HD and UHD resolutions with bit depth of 8 bits per pixel are used [60]. The test sequences and their details are summarized in Table 4-2. The standard test conditions and configurations given in [49] are used. The random access main configuration [49] is used for evaluation and the rate control parameters are enabled for encoding.

Figure 4-4 Sample frame from PeopleOnStreet Test Sequence

Figure 4-5 Sample frame from Sintel39 Test Sequence

Figure 4-6 Sample frame from ParkJoy Test Sequence

Table 4-2 Summary of test sequences

Test Sequence	Frame rate	HD Resolution	UHD Resolution	Total number of frames
PeopleOnStreet	30	1920x1080	3840x2160	150
Sintel39	24	1920x872	3840x1744	344
ParkJoy	50	1920x1080	3840x2160	500

4.5 Description of Experiments

Firstly, SHVC with 2 spatial layers having HD and UHD resolutions is encoded. Here the ratio of bits into base and enhancement layers called the bit allocation ratio (BL: EL) is kept at 50:50. For example, if the total bitrate is 8 Mbps, 4 Mbps is allocated for base layer and other 4 Mbps is allocated for the enhancement layer. Rate control feature present in the reference software is used to set the bitrate. The simulation is repeated for total bit rate of 8 Mbps to 40 Mbps in steps of 4 Mbps, for all the three test sequences using HEVC and MPEG-4 AVC as base layers. The bitrate savings is given by BD-Bitrate metric and scalability overhead is given by the BD-PSNR metric.

Secondly, SHVC with 3 layers having HD base layer, UHD enhancement layer 1 (i.e. 2 spatial layers) and one quality enhanced UHD enhancement layer 2 is encoded. The bit rate allocation ratio is kept at 50:25:25. For example, if total bitrate is 12Mbps, then 6 Mbps is allocated for BL, 3Mbps is allocated for EL1 and 3Mbps for EL2. The total bitrate is varied from 8 to 40 Mbps in steps of 4Mbps. The simulations are performed on PeopleOnStreet and Sintel39 test sequences with HEVC and MPEG-4 AVC as base layers and the bitrate savings and scalability overhead for 3 layers are obtained.

Further, the bitrate allocation ratio for 2 layers (spatial layers) of SHVC is varied from 10:90 to 90:10 in steps of 10. This simulation is performed on PeopleOnStreet test sequence and for HEVC and MPEG-4 AVC base layers. This is used to study the effect of ratio of bitrate allocation on bitrate savings and scalability overhead.

Additionally, the base layer bitrate is kept constant for a given total bitrate and enhancement layer bit rate is varied for 2 layers (spatial layers) of SHVC. For example, when BL bitrate is fixed at 4 Mbps, if total bitrate = 8Mbps, then EL = 4 Mbps; if total bitrate = 12 Mbps, then EL = 8Mbps. The simulation is repeated for total bitrate varying from 8 to 40 Mbps in steps of 4 Mbps. The base layer bitrate is fixed at 4, 6, 8, 10, 12, 16, 18 and 20 Mbps and these simulations are repeated. This simulation is performed for 2 layers of SHVC on PeopleOnStreet test sequence using HEVC as a base layer. This is used to study the effect of base layer bit allocation on scalability overhead in SHVC.

4.6 Summary

This chapter explains the need for SHVC evaluation in the context of HTTP streaming to deploy UHD. It describes about how R-D plots are obtained and also performance comparison metrics such as the PSNR and BD-metrics. The SHVC evaluation methodology is presented and experimental set-up is outlined. It also describes the different simulations performed for SHVC evaluation with 2 and 3 layers. The next chapter presents the results and discussion of SHVC evaluation.

Chapter 5

RESULTS AND DISCUSSION OF SHVC EVALUATION

The results of SHVC investigations are presented in this chapter. Bash scripts and python scripts are used for automating encoding process, extracting and plotting results.

5.1 Two layers of SHVC with bit allocation ratio of 50:50

The bitrate allocation ratio is fixed at 50:50, R-D plots are obtained for comparing SHVC with Simulcast and Single Layer Coding, with 2 layers having HD BL and UHD EL, using HEVC and MPEG-4 AVC as BL codecs. Figures 5-1 - 5-3 are the R-D plots obtained for PeopleOnStreet test sequence. Figures 5-4 - 5-6 are the R-D plots obtained for Sintel39 test sequence. Figures 5-7 - 5-9 are the R-D plots obtained for ParkJoy test sequence. The bitrate savings and the scalability overhead for 2 layers for all the test sequences are summarized in Figures 5-10 and 5-11.

5.1.1 Results for PeopleOnStreet

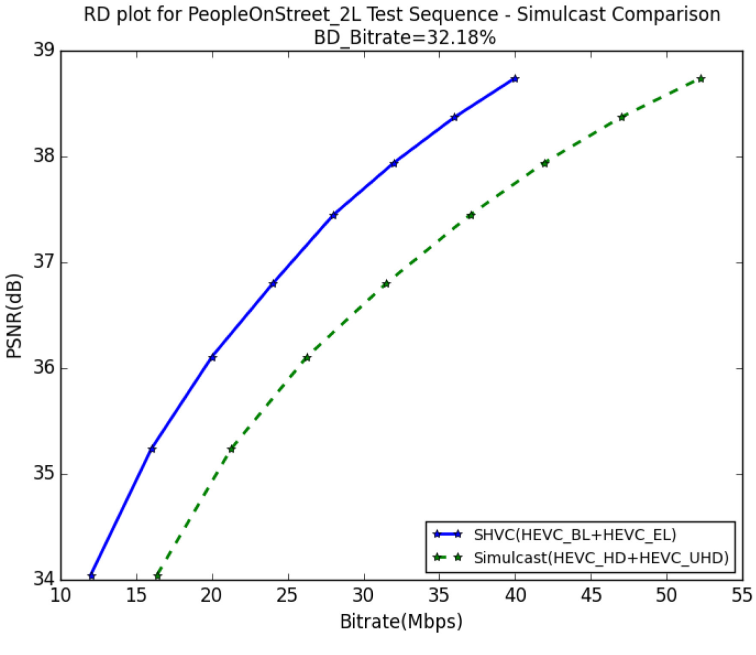


Figure 5-1 SHVC Vs Simulcast Coding Comparison, PeopleOnStreet, 2 layers, HEVC BL

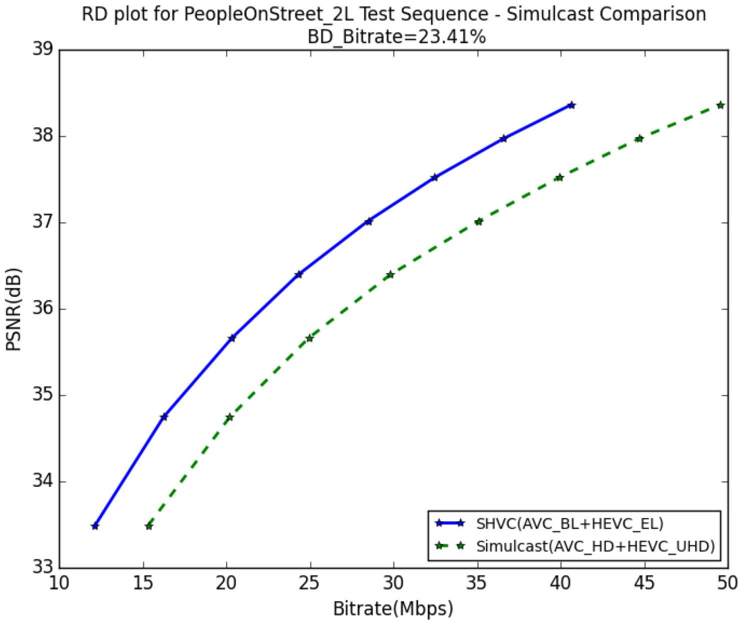


Figure 5-2 SHVC Vs Simulcast Coding Comparison, PeopleOnStreet, 2 layers, MPEG-4 AVC BL

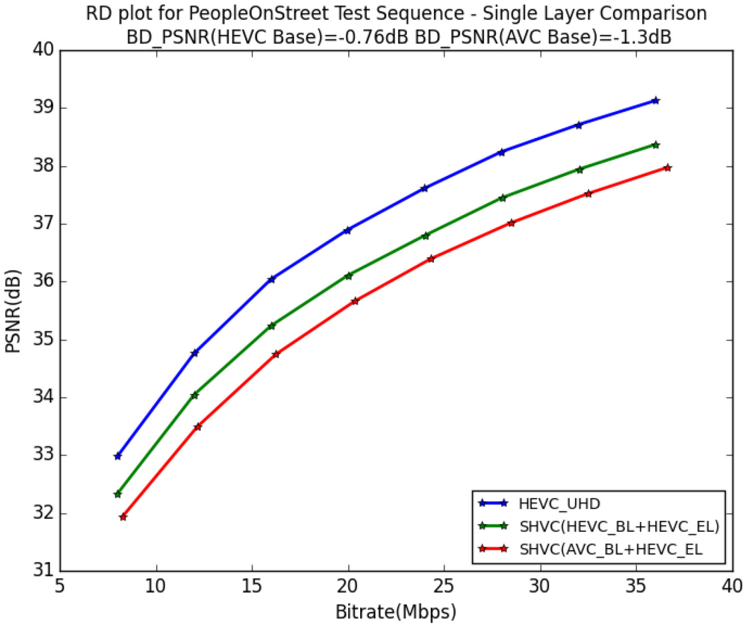


Figure 5-3 SHVC Vs Single Layer Coding Comparison, PeopleOnStreet, 2 layers, HEVC and MPEG-4 AVC BL

5.1.2 Results for Sintel39

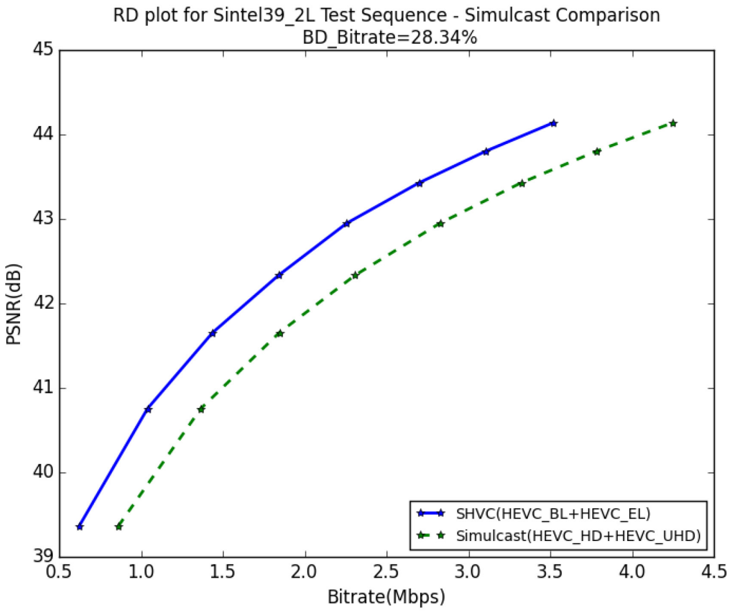


Figure 5-4 SHVC Vs Simulcast Coding Comparison, Sintel39, 2 layers, HEVC BL

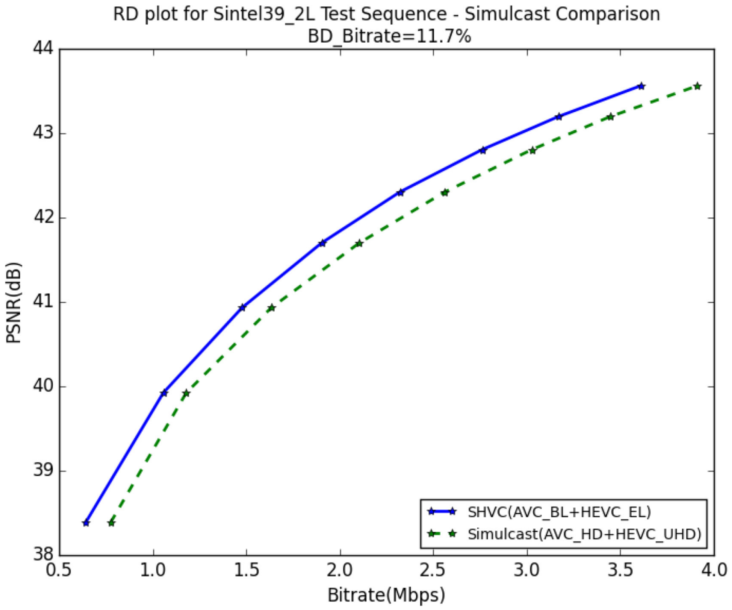


Figure 5-5 SHVC Vs Simulcast Coding Comparison, Sintel39, 2 layers, MPEG-4 AVC BL

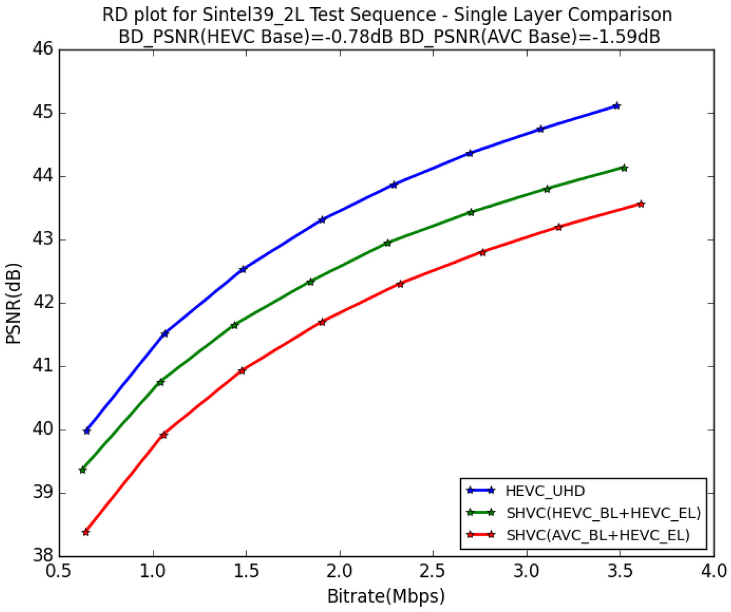


Figure 5-6 SHVC Vs Single Layer Coding Comparison, Sintel39, 2 layers,

HEVC and MPEG-4 AVC BL

5.1.3 Results for ParkJoy

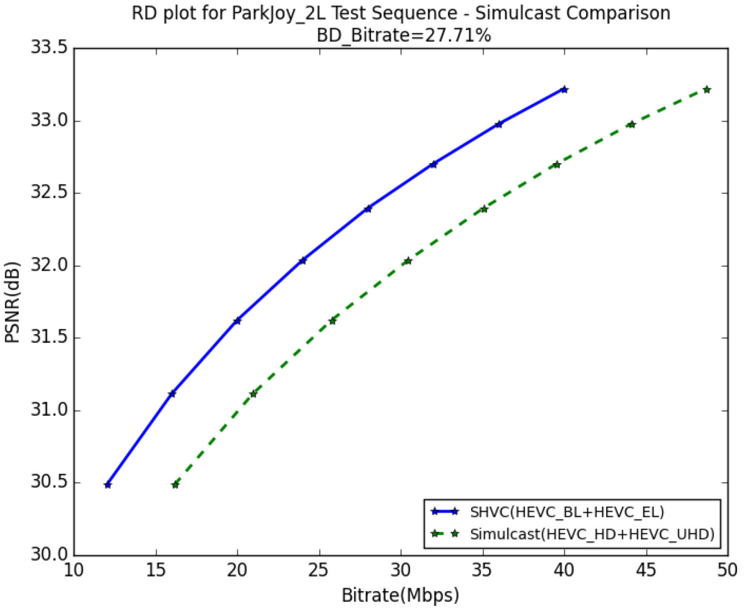


Figure 5-7 SHVC Vs Simulcast Coding Comparison, ParkJoy, 2 layers, HEVC BL

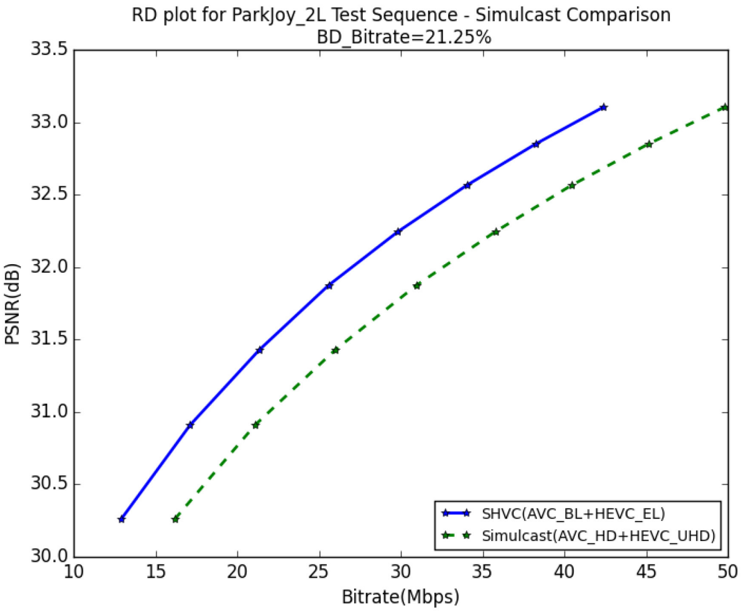


Figure 5-8 SHVC Vs Simulcast Coding Comparison, ParkJoy, 2 layers, MPEG-4 AVC BL

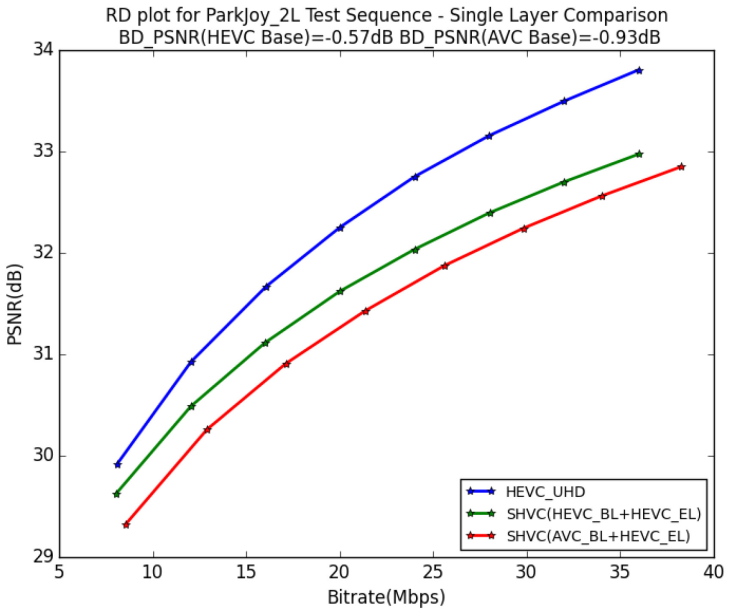


Figure 5-9 SHVC Vs Single Layer Coding Comparison, ParkJoy, 2 Layers, HEVC and MPEG-4 AVC BL

5.1.4 Summary of results for 2 layers

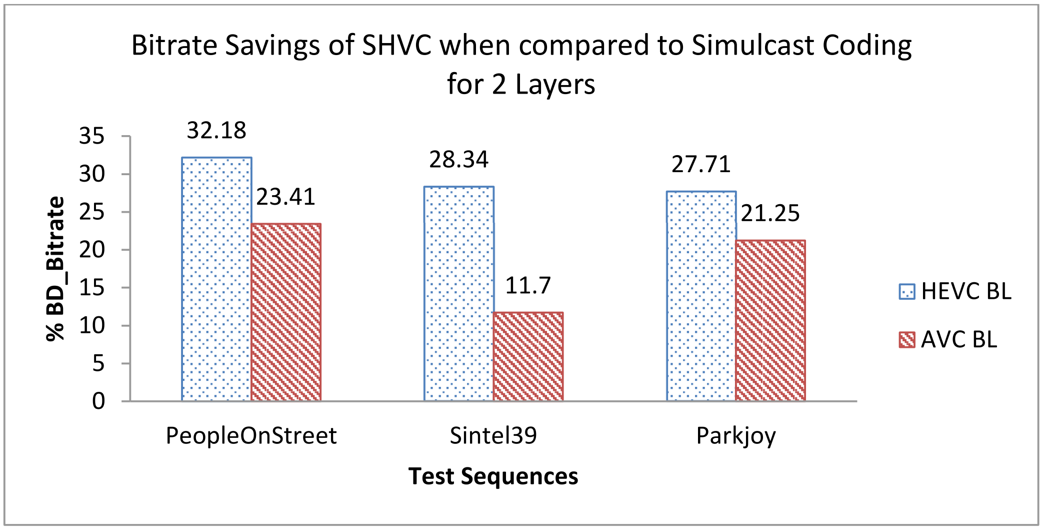


Figure 5-10 Summary of Bitrate savings for SHVC Vs Simulcast Coding, 2 Layers, HD base layer and UHD enhancement layer

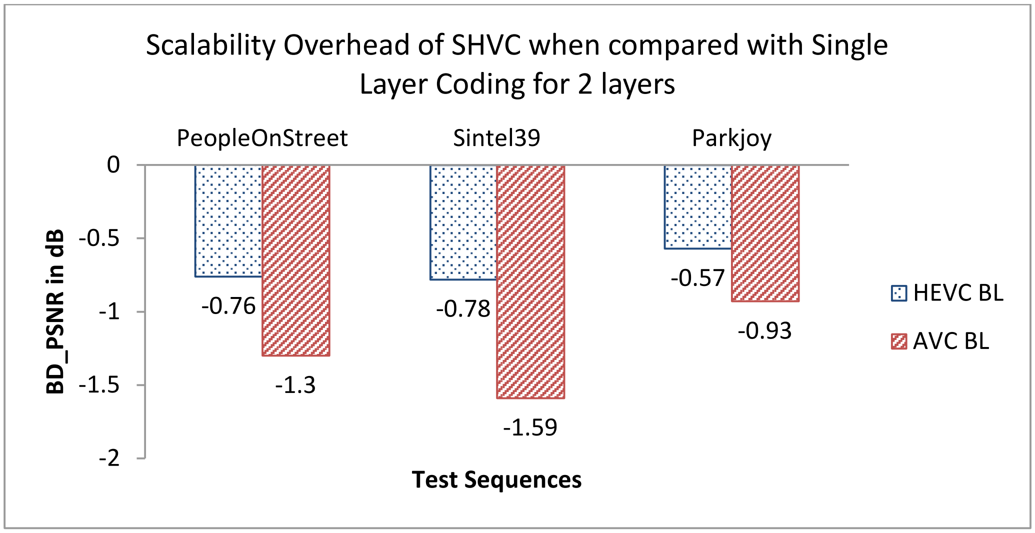


Figure 5-11 Summary of Scalability Overhead of SHVC when compared to Single Layer Coding, 2 Layers, HD base layer and UHD enhancement layer

5.2 Three layers of SHVC with bit allocation ratio of 50:25:25

The bitrate allocation ratio is fixed at 50:25:25, R-D plots are used for comparing SHVC with Simulcast and Single Layer Coding, with 3 layers having HD BL, UHD EL1 and quality enhanced UHD EL2, using HEVC and MPEG-4 AVC as BL codecs. Figure 5-12 and Figure 5-13 represent the R-D plot for PeopleOnStreet and Sintel39 test sequences, used for obtaining the scalability overhead. The bitrate savings and the scalability overhead for 3 layers for PeopleOnStreet and Sintel39 test sequences are summarized in Figure 5-14 and Figure 5-15.

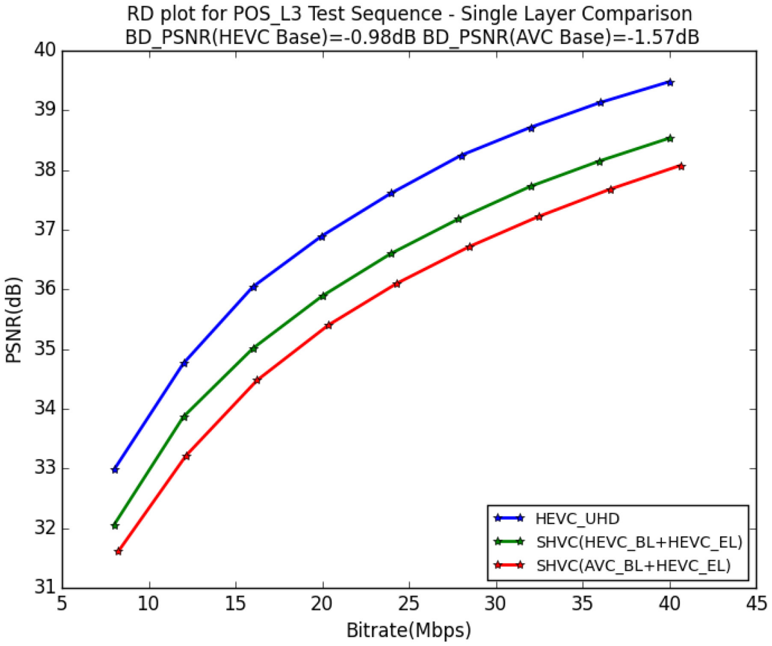


Figure 5-12 Comparison of SHVC Vs Single Layer Coding, BL:EL1:EL2 = 50:25:25, HEVC and MPEG-4 AVC BL, 3 layers, PeopleOnStreet

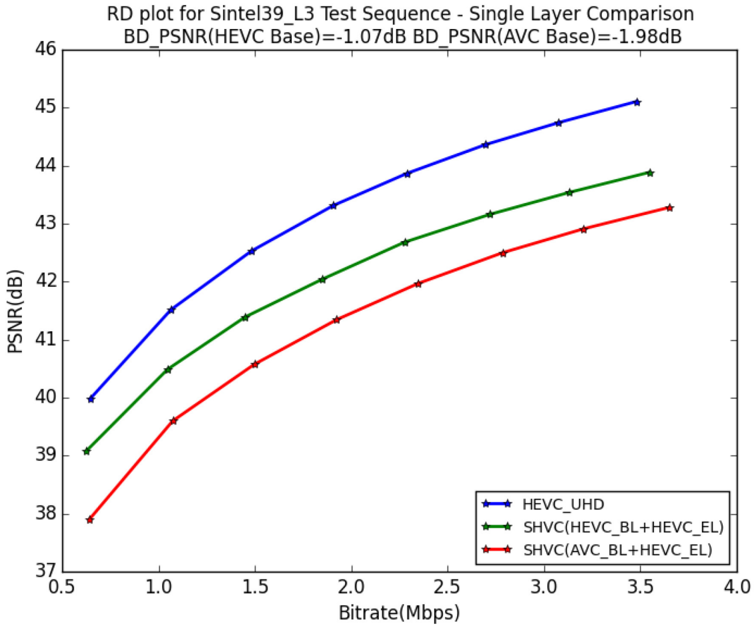


Figure 5-13 Comparison of SHVC Vs Single Layer Coding, BL:EL1:EL2=50:25:25, HEVC and MPEG-4 AVC BL, 3 layers, Sintel39

5.2.1 Summary of results for 3 layers

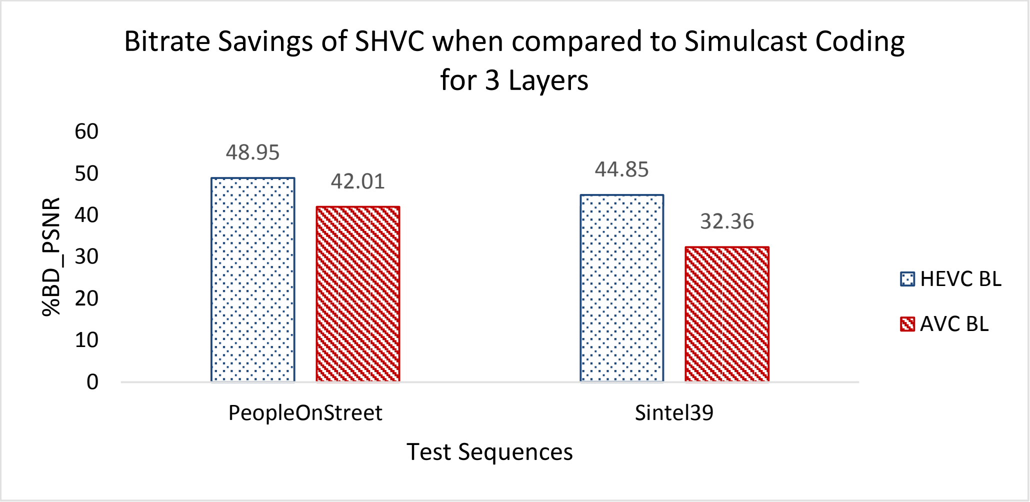


Figure 5-14 Summary of Bitrate savings for SHVC Vs Simulcast Coding, 3 Layers, HD BL, UHD EL1 and Quality enhanced UHD EL2

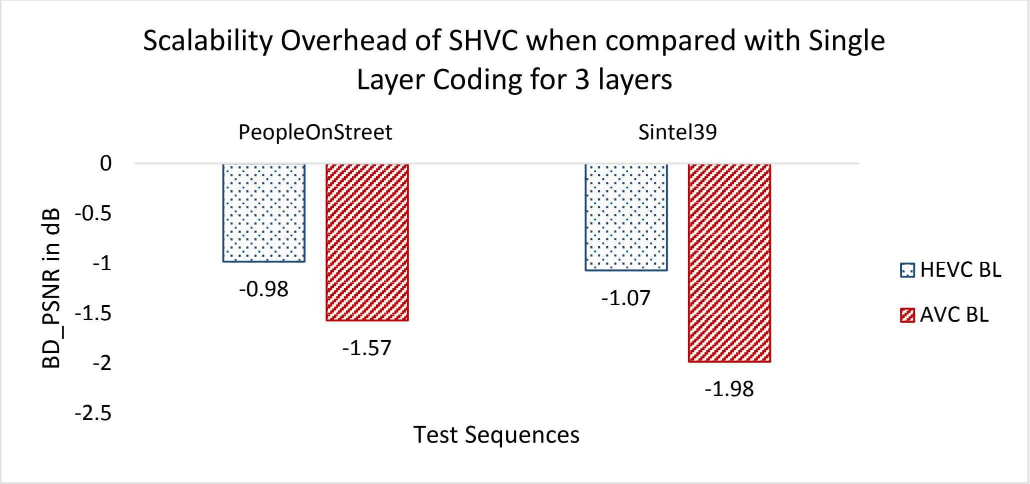


Figure 5-15 Summary of Scalability Overhead for SHVC when compared to Single Layer Coding, 3 Layers, HD BL, UHD EL1 and Quality enhanced UHD EL2

5.3 Varying ratio of bit allocation for 2 layers

The bitrate allocation ratio BL: EL is varied from 10:90 to 90:10 in steps of 10 and R-D plots are obtained for comparing SHVC with Simulcast and Single Layer Coding; with 2 layers having HD BL & UHD EL, using HEVC and MPEG-4 AVC as BL codecs for PeopleOnStreet test sequence in Figures 5-16 - 5-39. The bitrate savings and the scalability overhead for 2 layers for PeopleOnStreet test sequence are summarized in Figures 5-40 and 5-41.

5.3.1 Comparison of SHVC Vs Simulcast Coding

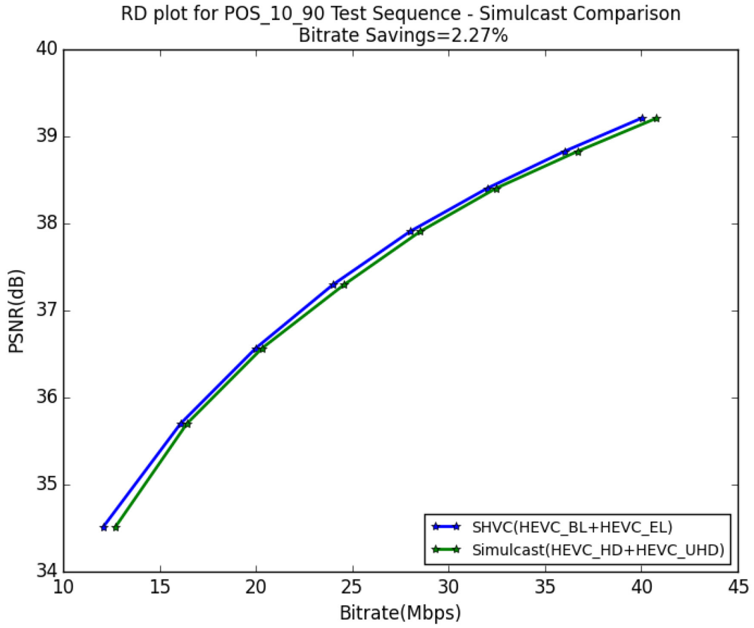


Figure 5-16 SHVC Vs Simulcast Coding, BL:EL =10:90, HEVC BL

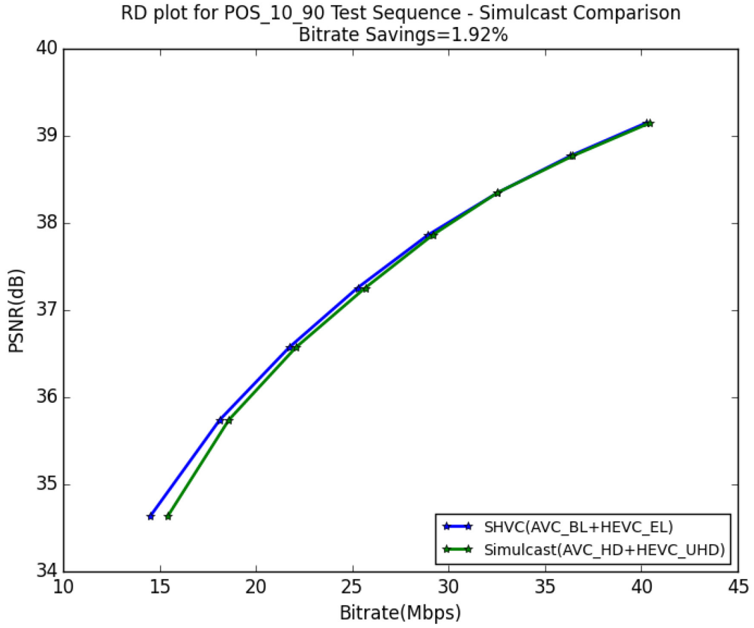


Figure 5-17 SHVC Vs Simulcast Coding, BL:EL =10:90, MPEG-4 AVC BL

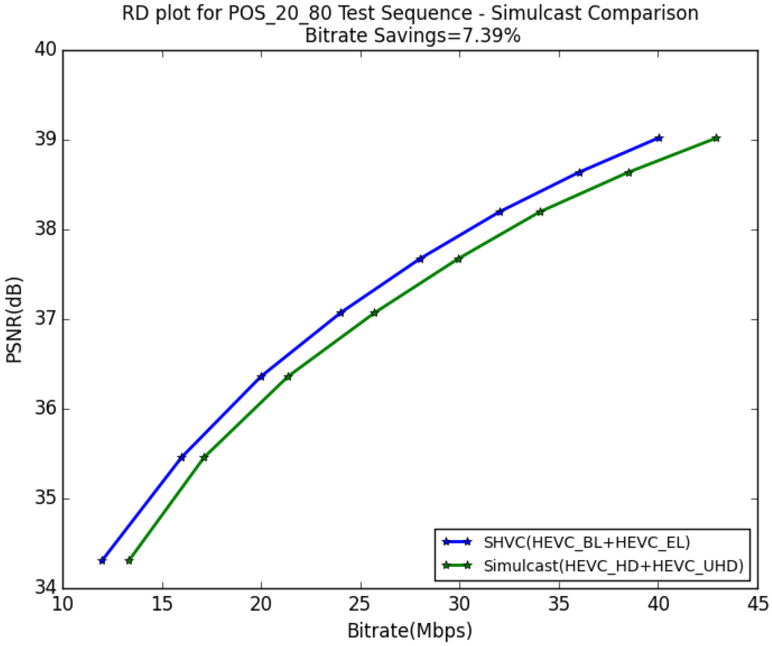


Figure 5-18 SHVC Vs Simulcast Coding, BL:EL =20:80, HEVC BL

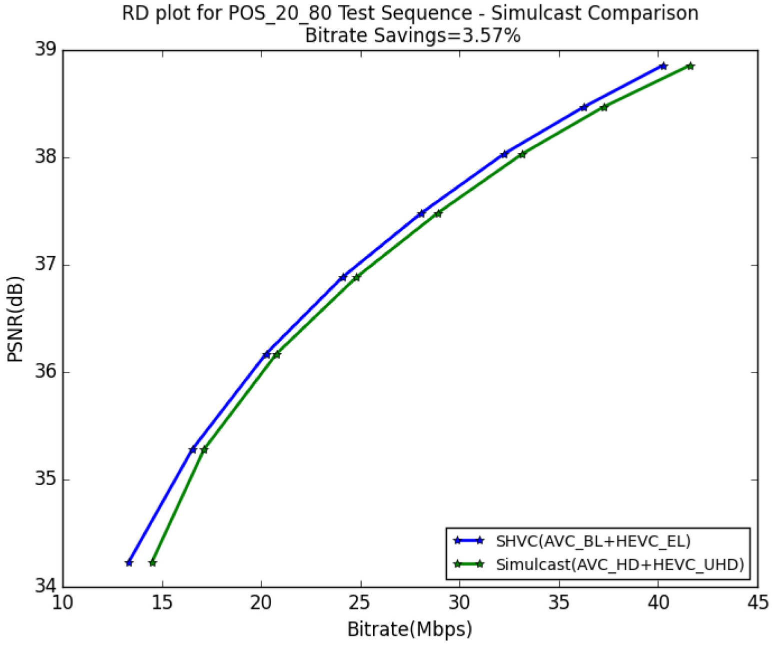


Figure 5-19 SHVC Vs Simulcast Coding, BL:EL =20:80, MPEG-4 AVC BL

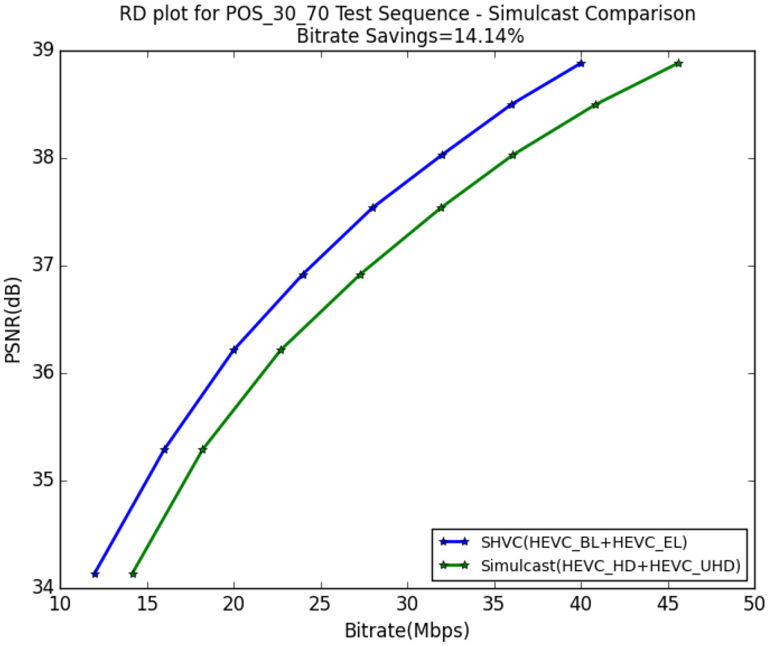


Figure 5-20 SHVC Vs Simulcast Coding, BL:EL =30:70, HEVC BL

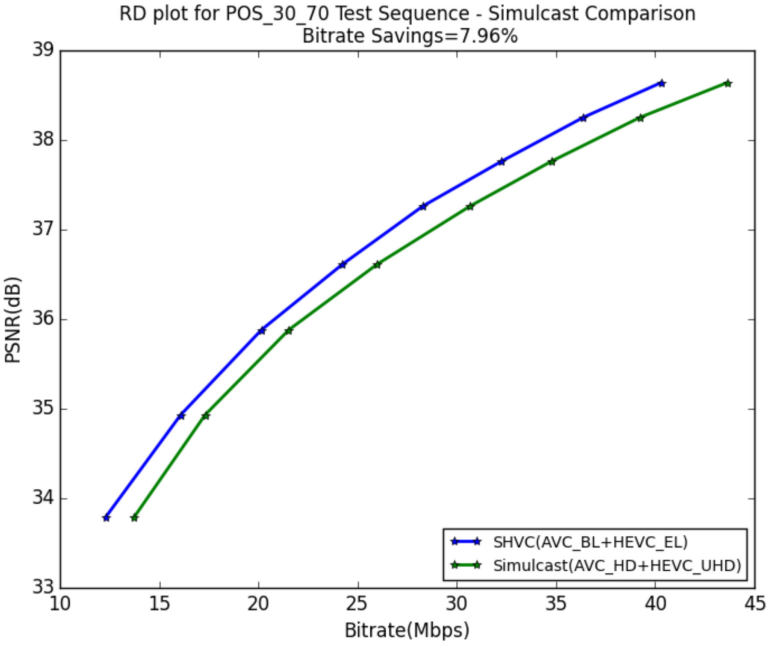


Figure 5-21 SHVC Vs Simulcast Coding, BL:EL = 30:70, MPEG-4 AVC BL

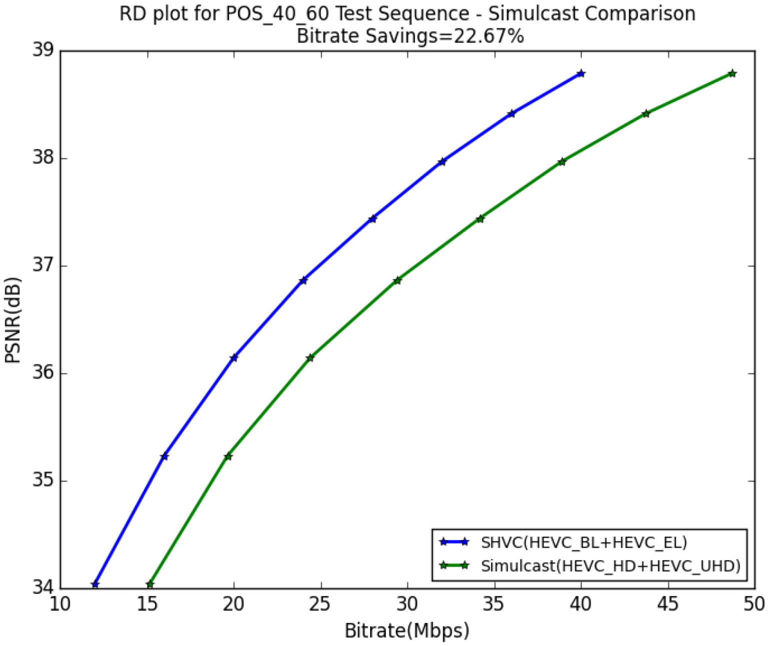


Figure 5-22 SHVC Vs Simulcast Coding, BL:EL =40:60, HEVC BL

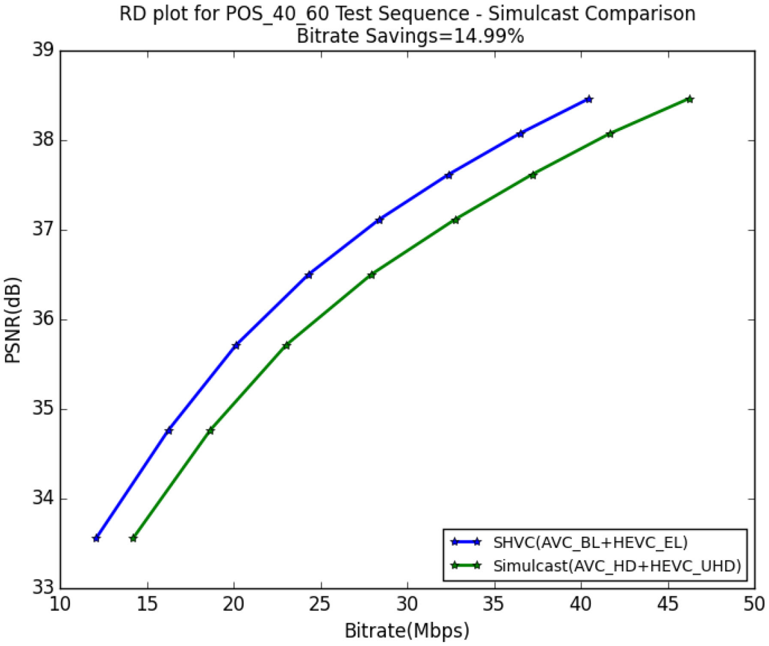


Figure 5-23 SHVC Vs Simulcast Coding, BL:EL = 40:60, MPEG-4 AVC BL

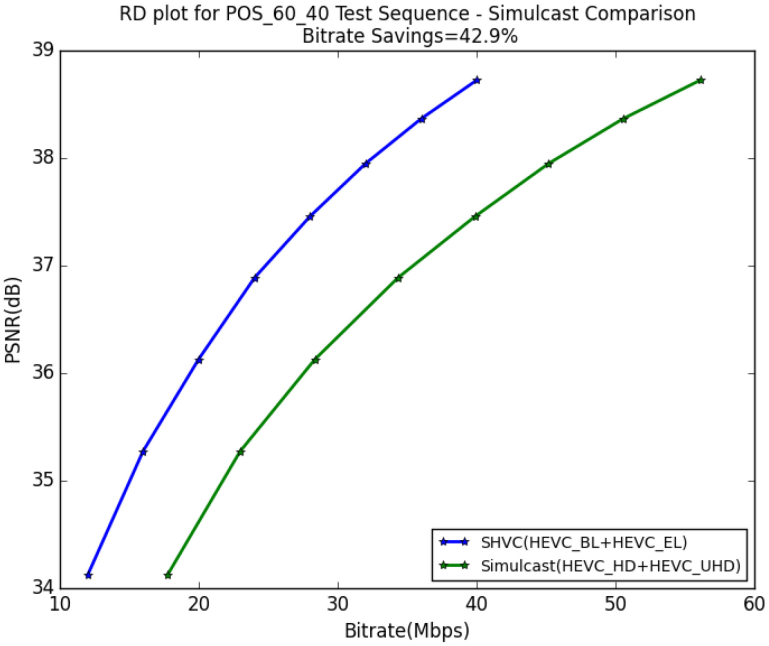


Figure 5-24 SHVC Vs Simulcast Coding, BL:EL =60:40, HEVC BL

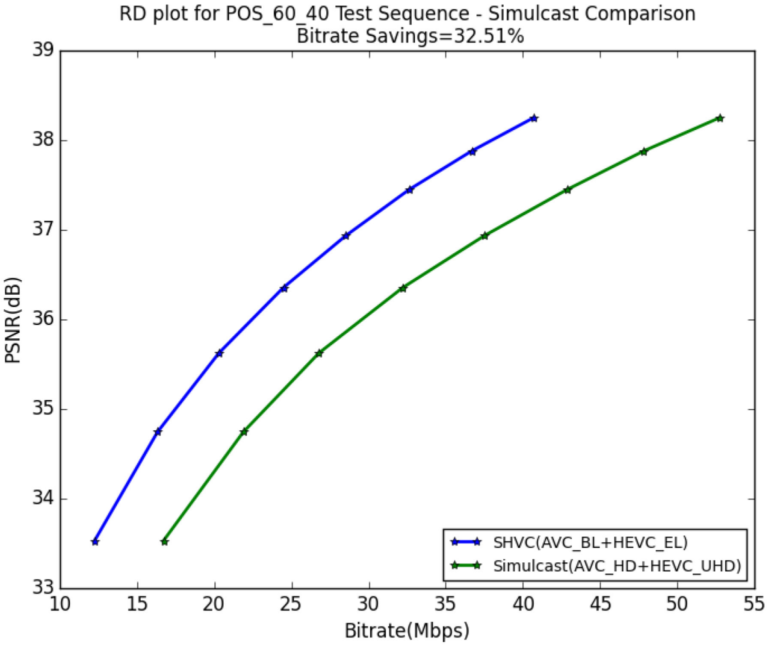


Figure 5-25 SHVC Vs Simulcast Coding, BL:EL = 60:40, MPEG-4 AVC BL

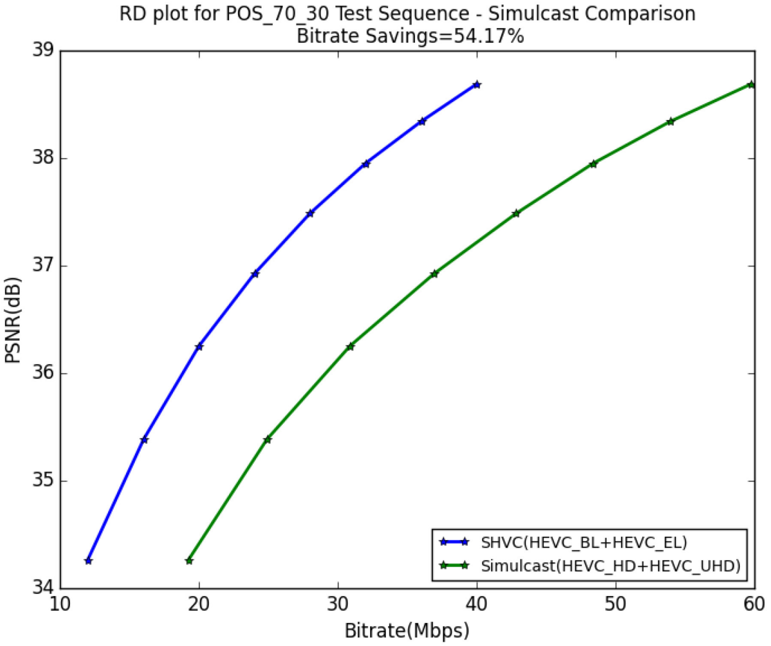


Figure 5-26 SHVC Vs Simulcast Coding, BL:EL =70:30, HEVC BL

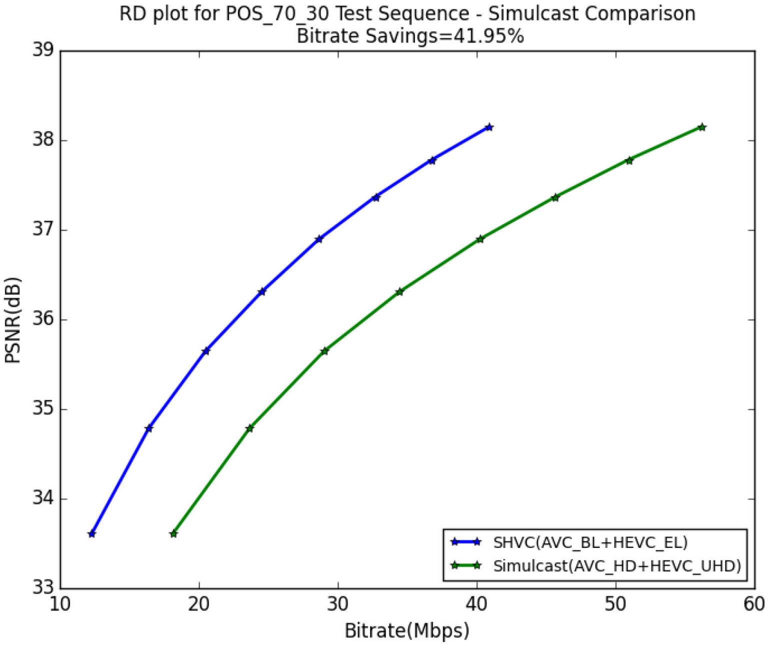


Figure 5-27 SHVC Vs Simulcast Coding, BL:EL = 70:30, MPEG-4 AVC BL

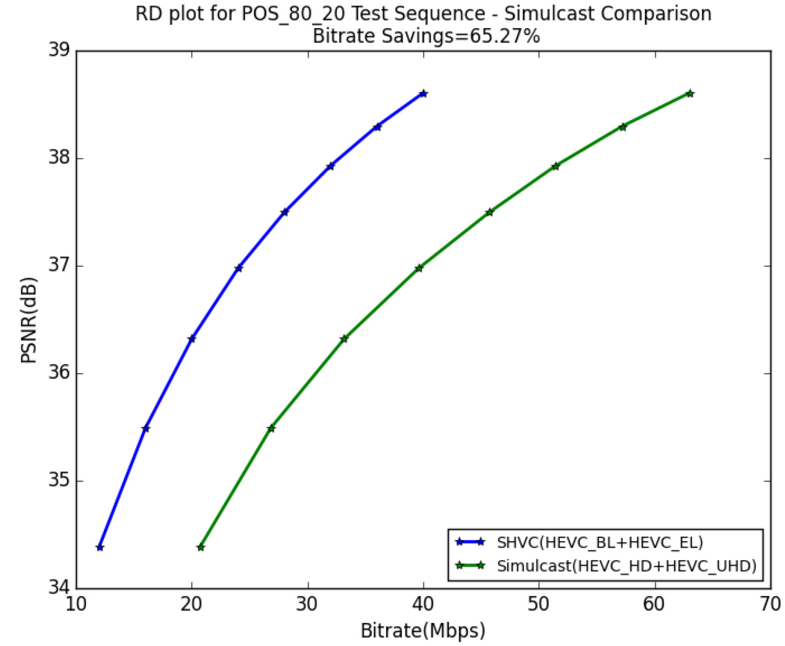


Figure 5-28 SHVC Vs Simulcast Coding, BL:EL =80:20, HEVC BL

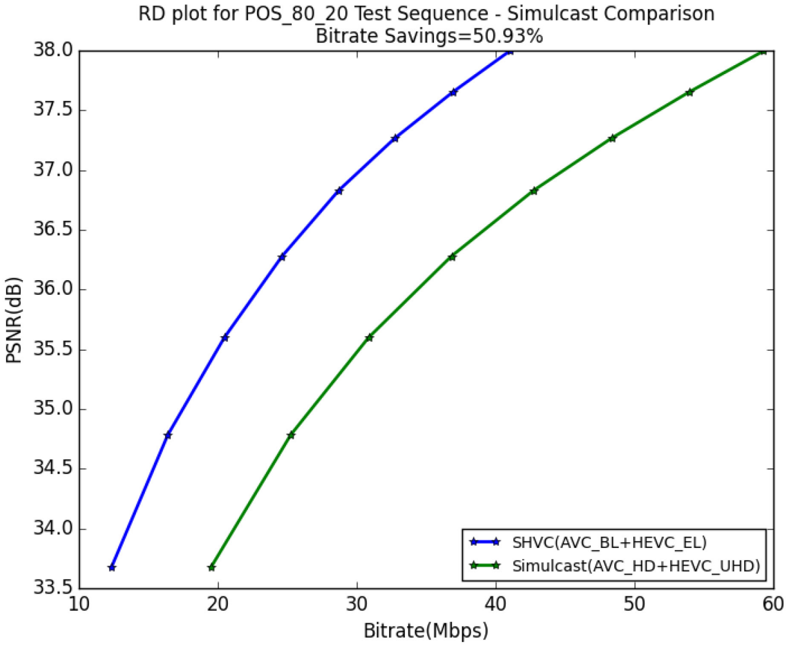


Figure 5-29 SHVC Vs Simulcast Coding, BL:EL = 80:20, MPEG-4 AVC BL

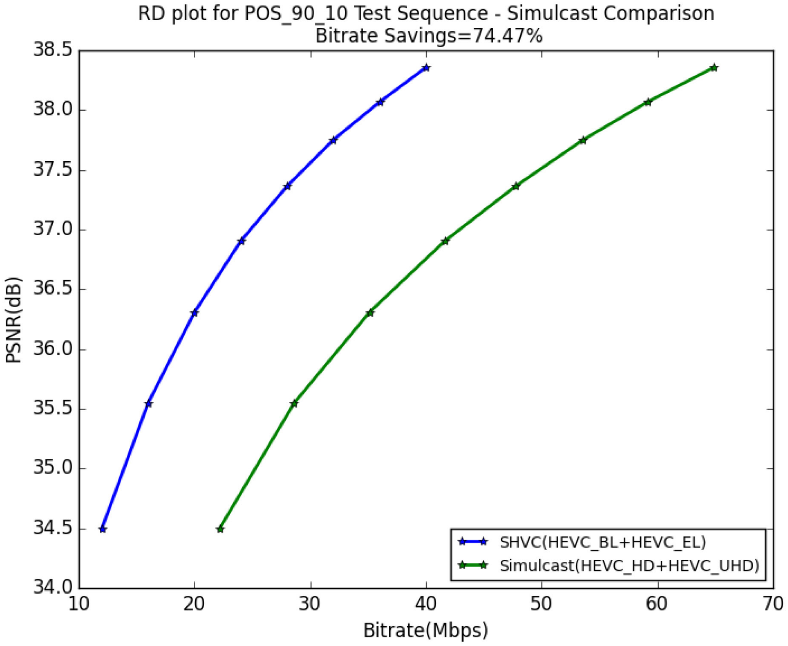


Figure 5-30 SHVC Vs Simulcast Coding, BL:EL =90:10, HEVC BL

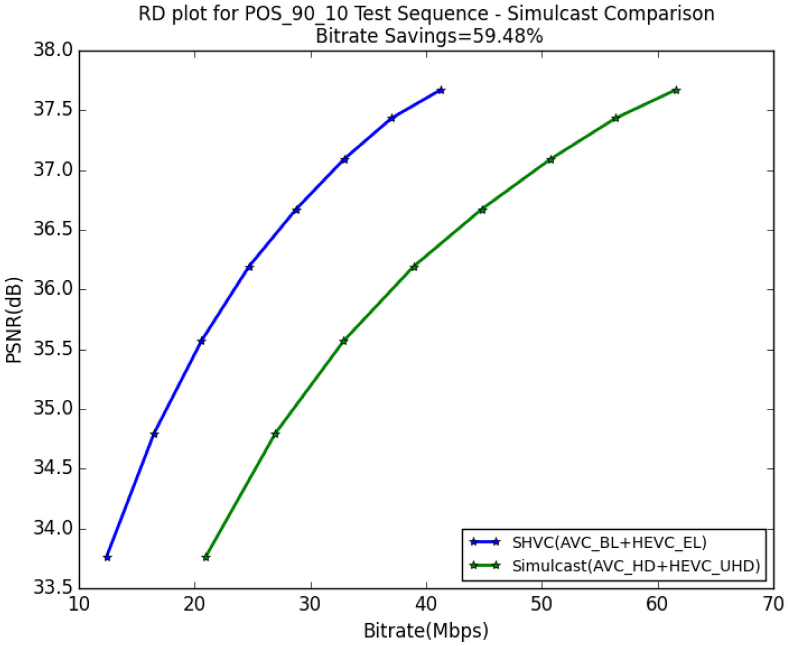


Figure 5-31 SHVC Vs Simulcast Coding, BL:EL = 90:10, MPEG-4 AVC BL

5.3.2 Comparison of SHVC Vs Single Layer Coding

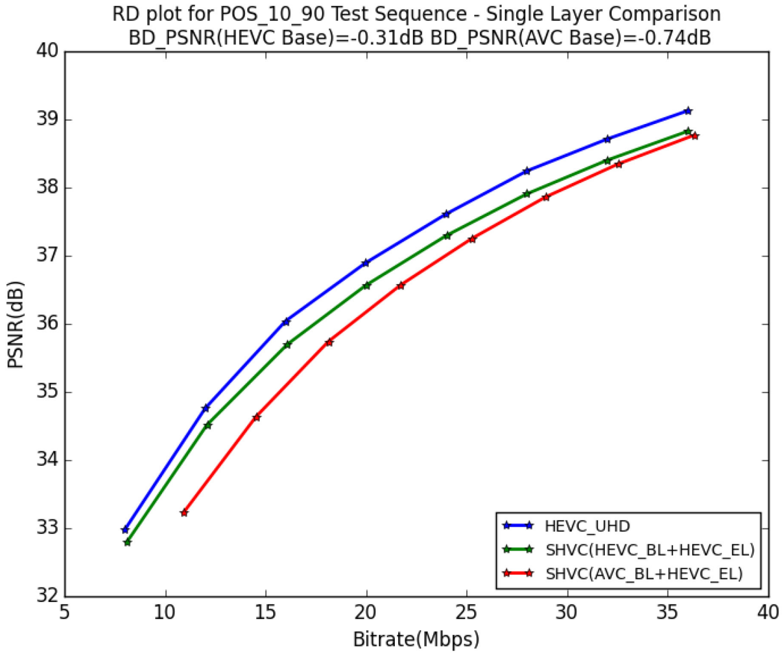


Figure 5-32 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 10:90

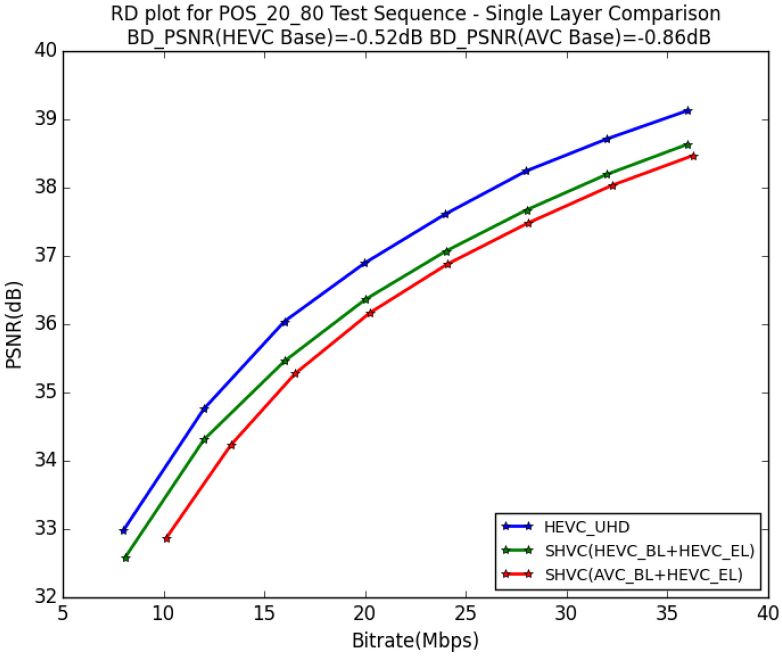


Figure 5-33 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 20:80

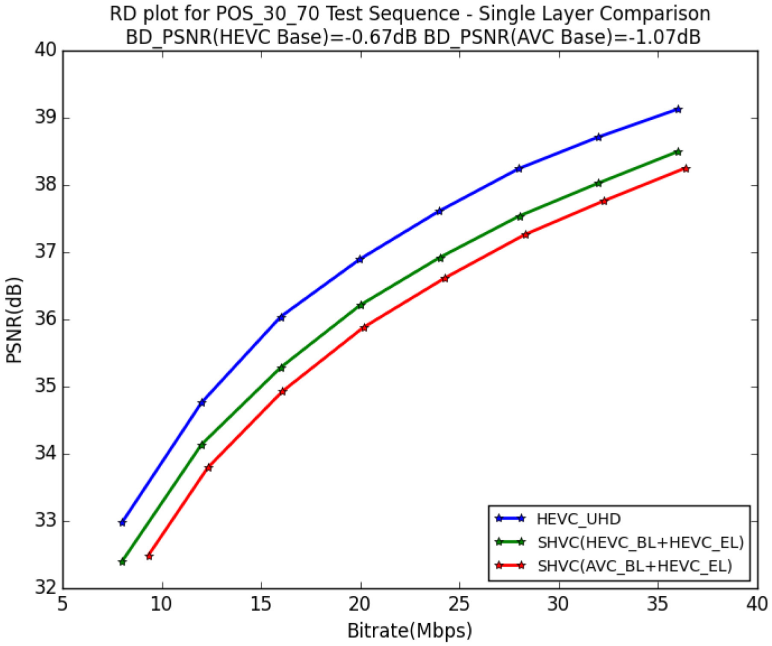


Figure 5-34 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 30:70

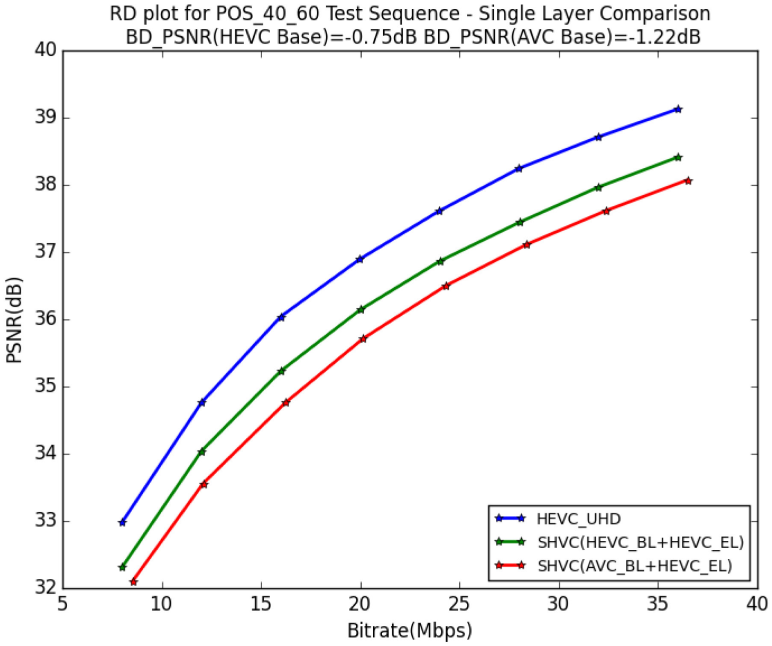


Figure 5-35 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 40:60

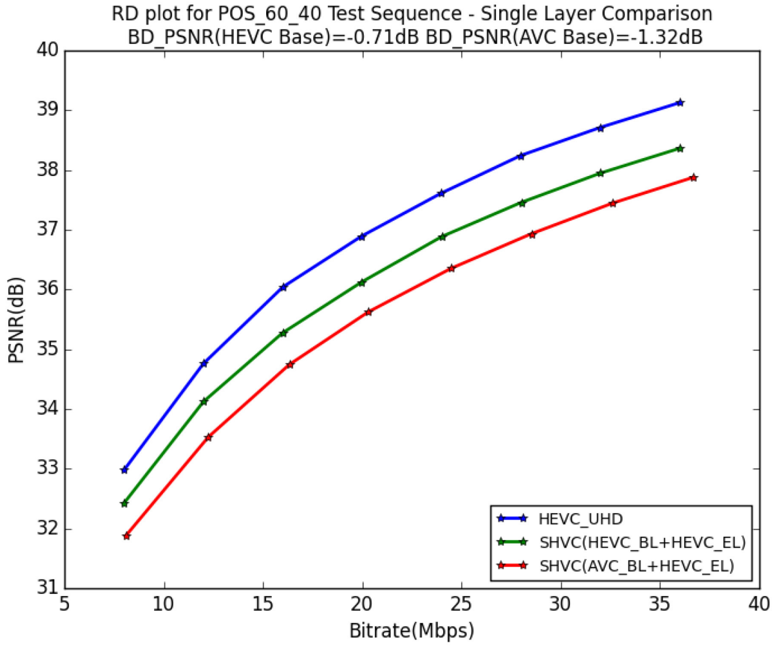


Figure 5-36 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 60:40

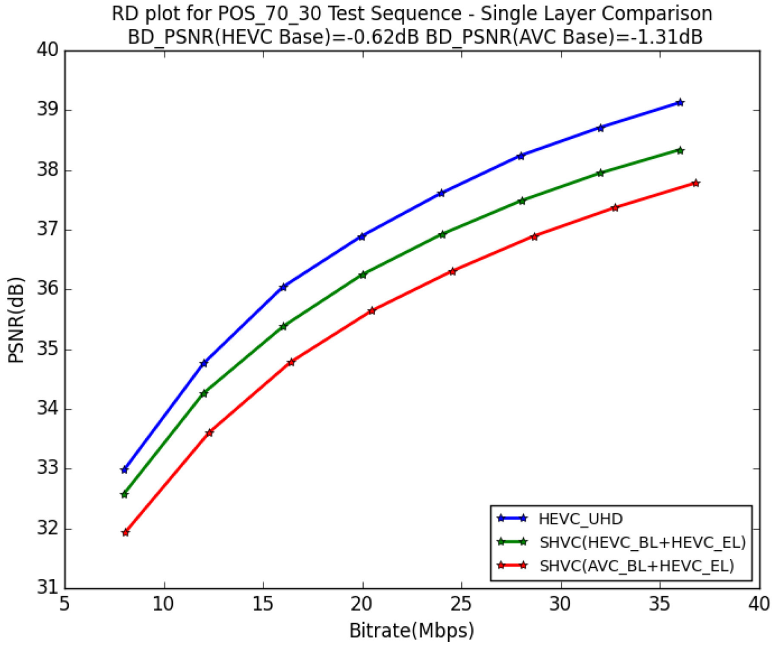


Figure 5-37 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 70:30

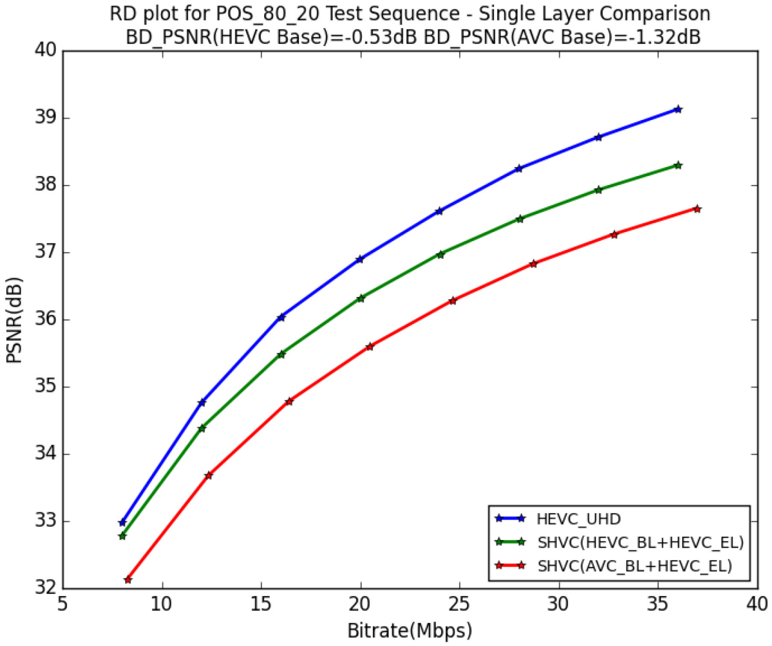


Figure 5-38 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 80:20

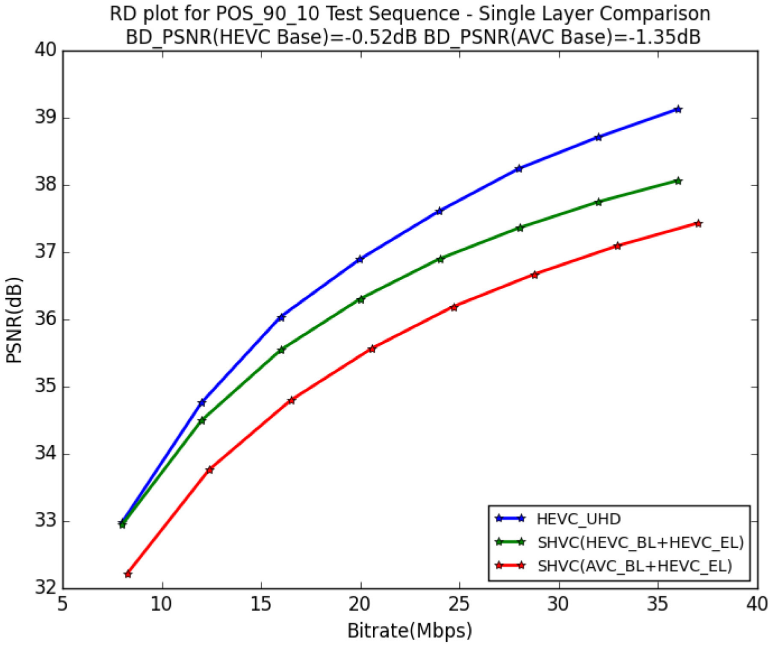


Figure 5-39 SHVC Vs Single Layer Coding, HEVC and MPEG-4 AVC BL, 90:10

5.3.3 Summary of results

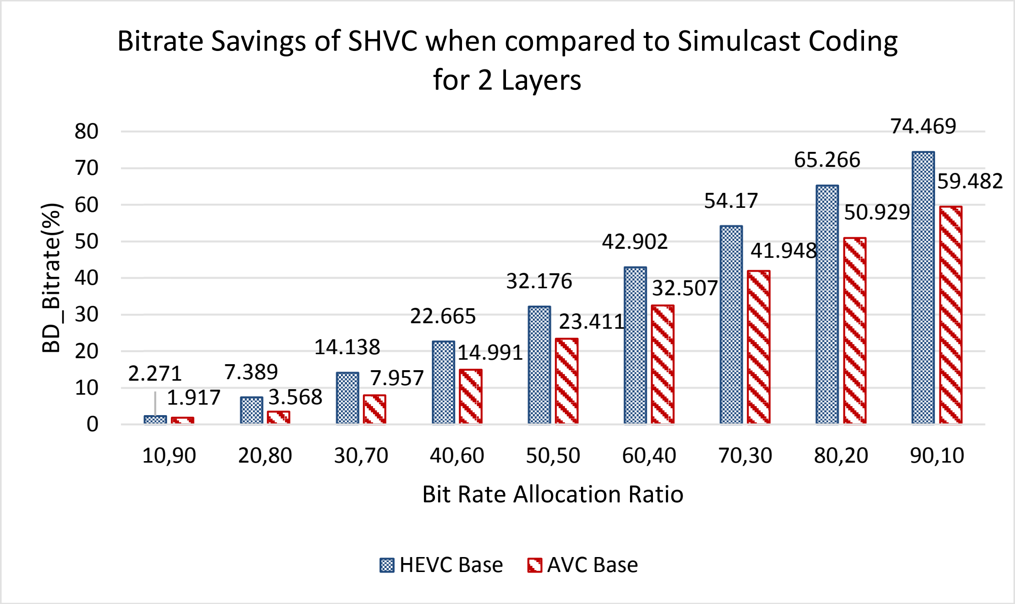


Figure 5-40 Summary of Bitrate savings for SHVC Vs Simulcast Coding, 2 Layers, HD BL and UHD EL by varying bit rate allocation ratio

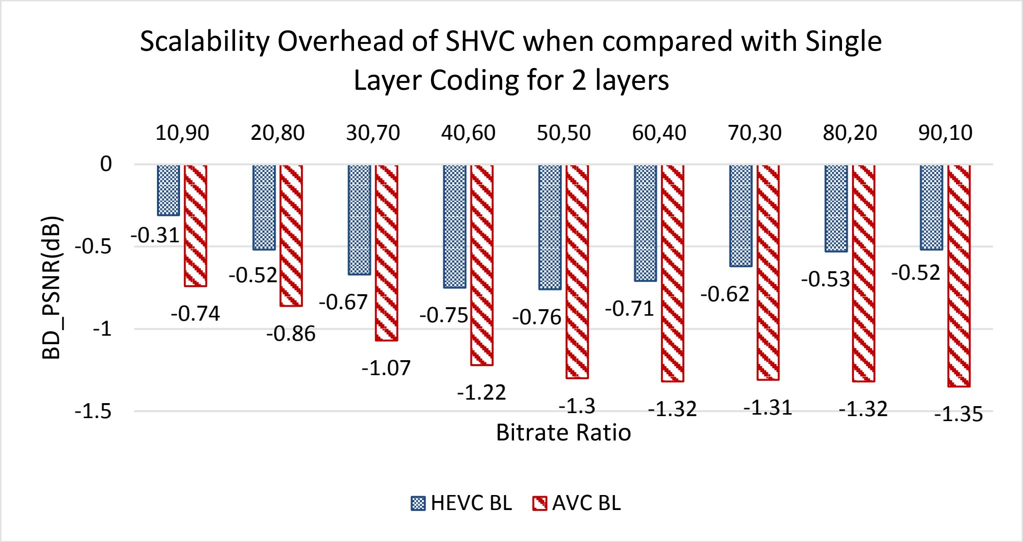


Figure 5-41 Summary of Scalability Overhead of SHVC when compared to Single Layer Coding, 2 Layers, HD BL and UHD EL, varying the bit rate allocation ratio

5.4 Fixed BL and varying EL bit allocation for 2 layers

Figure 5-42 through Figure 5-50 represent the R-D plots obtained by fixing the base layer bitrate at 4, 6, 8, 10, 12, 14, 16, 18 and 20 Mbps, having 2 layers, HD BL and UHD EL for PeopleOnStreet test sequence. The BL codec used here is HEVC. These R-D plots are used to obtain the scalability overhead by comparing SHVC with Single Layer Coding. The effect of base layer bit rate on scalability is summarized in Figure 5-51.

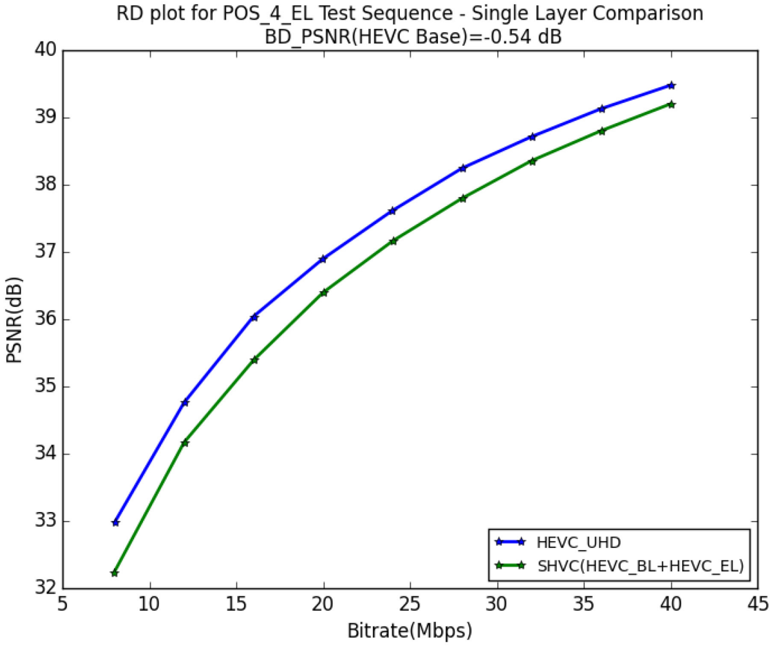


Figure 5-42 SHVC Vs Single layer Coding, HEVC BL, BL = 4 Mbps

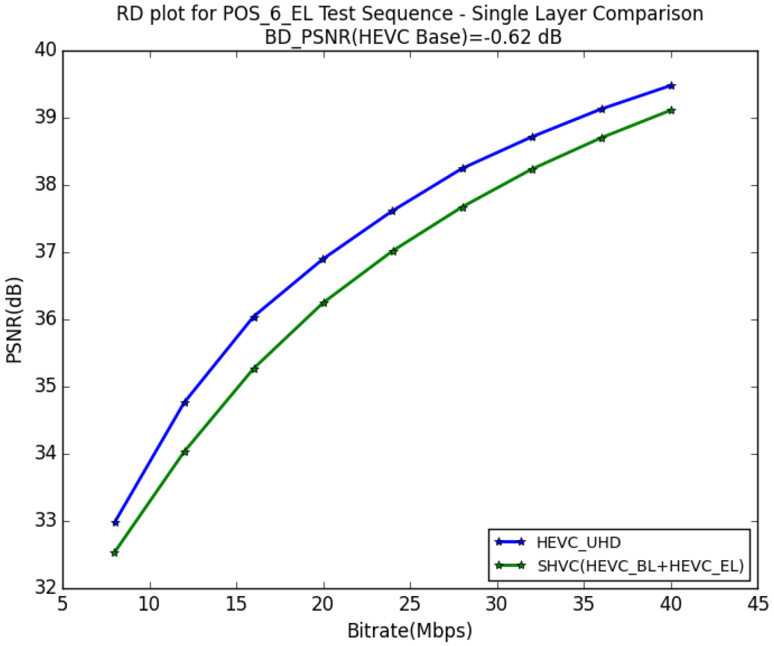


Figure 5-43 SHVC Vs Single layer Coding, HEVC BL, BL = 6 Mbps

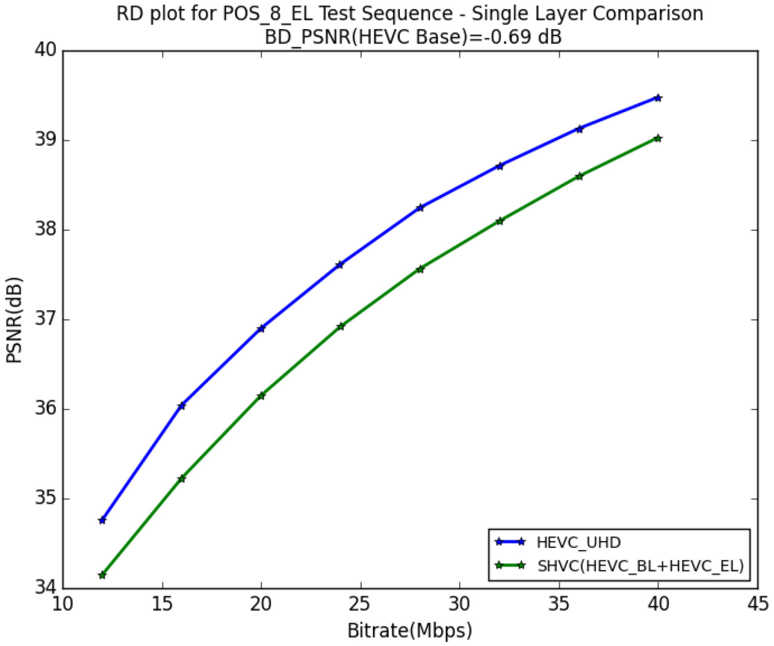


Figure 5-44 SHVC Vs Single layer Coding, HEVC BL, BL = 8 Mbps

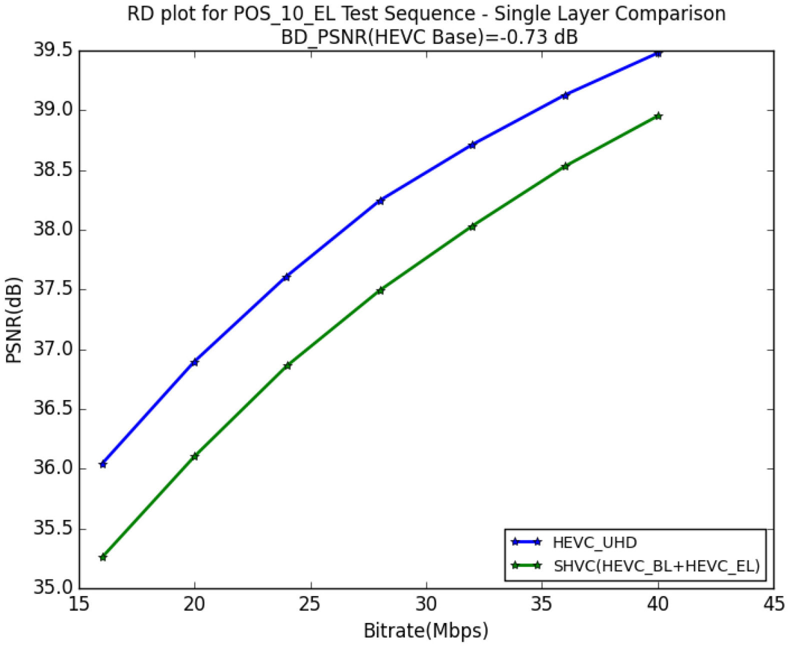


Figure 5-45 SHVC Vs Single layer Coding, HEVC BL, BL = 10 Mbps

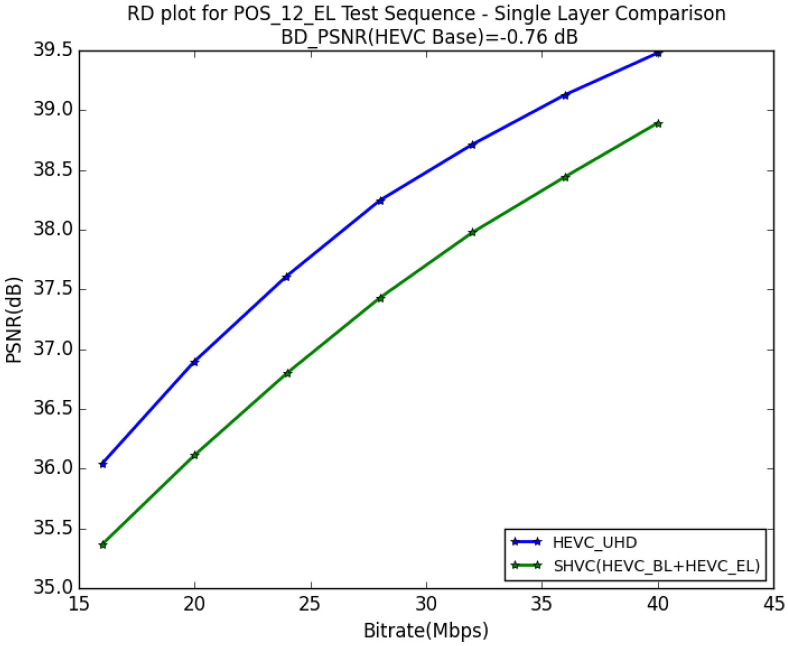


Figure 5-46 SHVC Vs Single layer Coding, HEVC BL, BL = 12 Mbps

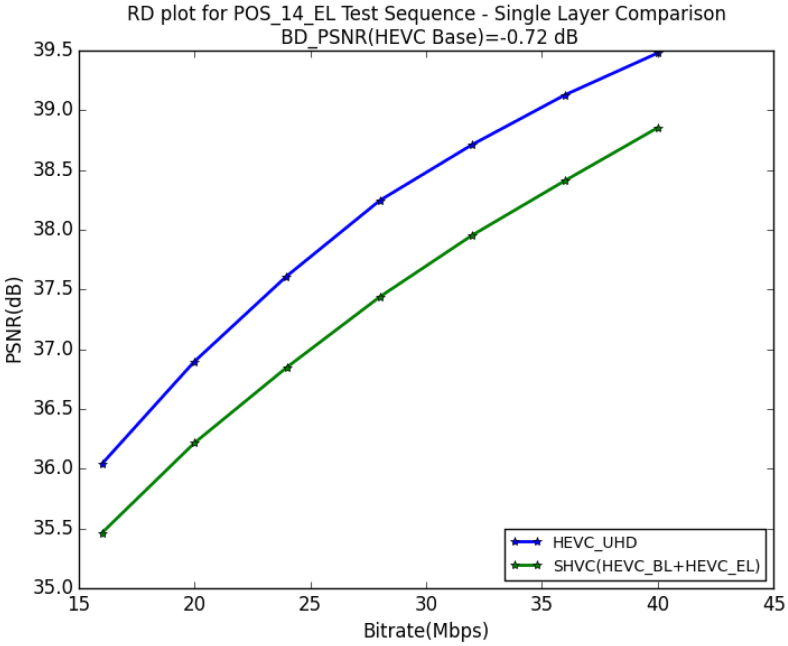


Figure 5-47 SHVC Vs Single layer Coding, HEVC BL, BL = 14 Mbps

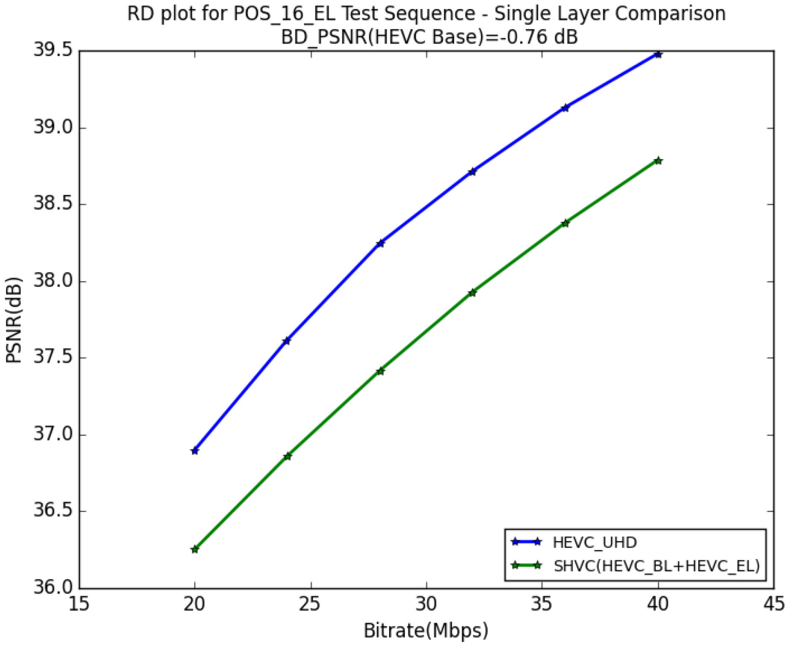


Figure 5-48 SHVC Vs Single layer Coding, HEVC BL, BL = 16 Mbps

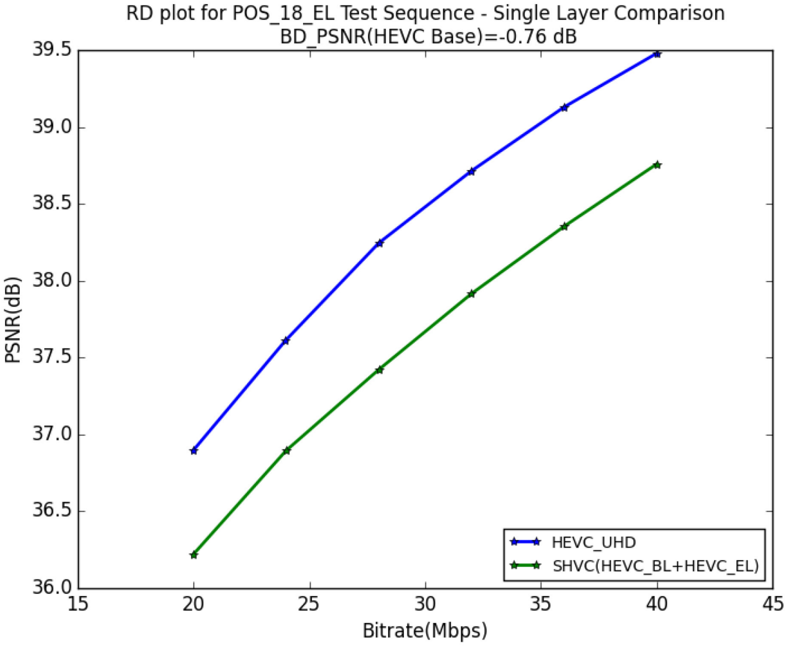


Figure 5-49 SHVC Vs Single layer Coding, HEVC BL, BL = 18 Mbps

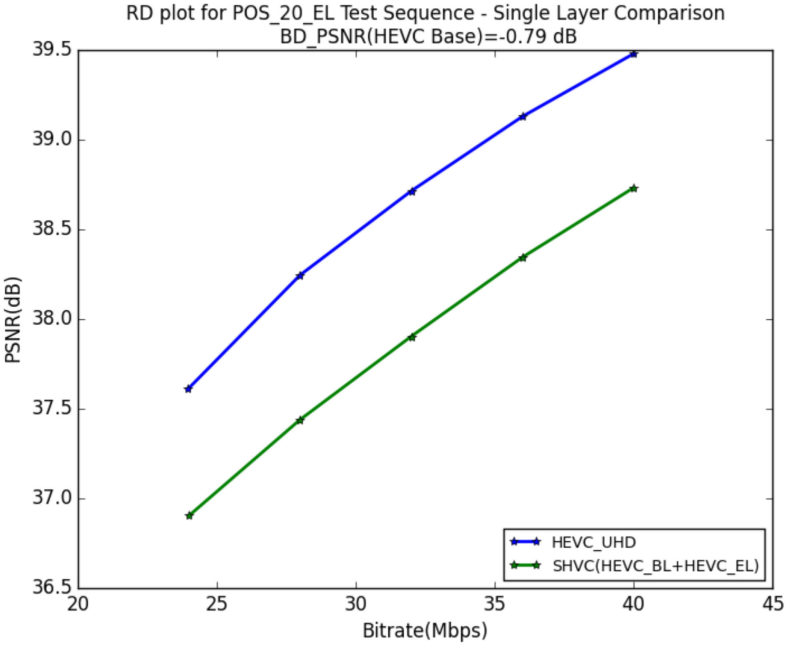


Figure 5-50 SHVC Vs Single layer Coding, HEVC BL, BL = 20 Mbps

5.4.1 Summary of results

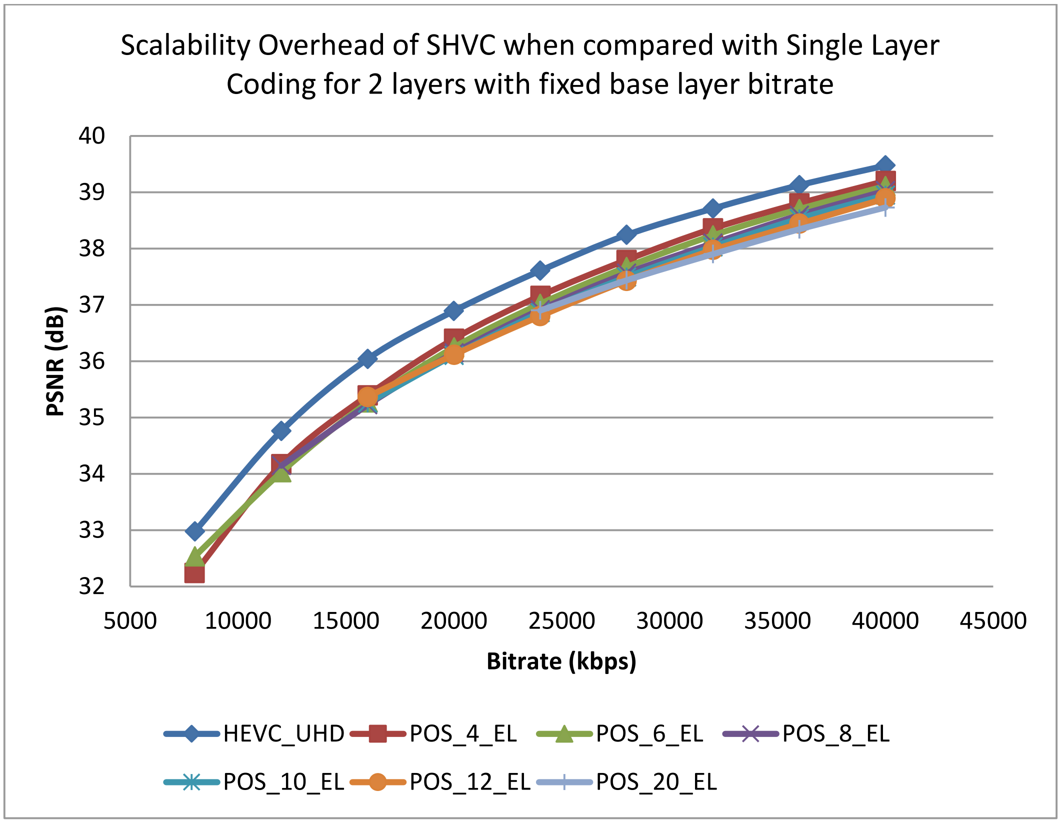


Figure 5-51 Effect of fixed base layer bitrate on SHVC Scalability Overhead

5.5 Discussion

From Figure 5-10 and Figure 5-11, it can be seen that for UHD deployment with two spatial layers of SHVC having HEVC as base layer resulted in 27%-32% bitrate savings and an average scalability overhead of 0.7 dB. Similarly, SHVC with MPEG-4 AVC as base layer resulted in 11%-23% bitrate savings and an average scalability overhead of 1.27 dB. As seen from Figure 5-14 and Figure 5-15, for combined spatial and quality scalability with 3 layers, using HEVC as base layer resulted in bitrate savings of 44%-49% and an average scalability overhead of 1.02 dB. Similarly, using MPEG-4 AVC as base layer resulted in 32%-42% bitrate savings and an average scalability overhead of 1.72 dB.

These results indicate that SHVC with HEVC as base layer performs better than compared to using MPEG-4 AVC as base layer.

As the number of SHVC layers are increased, better bitrate savings are obtained. However, the scalability overhead also increases. The number of scalable layers have to be decided considering both the bitrate savings and scalability overhead. From Figure 5-40 and Figure 5-41, it can be seen that the bit rate savings and scalability overhead depend on the ratio of bit rate allocation into various layers. Figure 5-51 indicates that for a given total bitrate, the scalability overhead in case of spatial scalability increases with increasing the bits in the base layer in SHVC. Increasing the bits allocated for the base layer does not necessarily increase the quality of the resulting video. The quality is impacted by the resolutions of the layers as well. Hence, there is a need to determine the optimal bit allocation for the scalable layers.

5.6 Summary

This chapter outlined the results and discussions on evaluation of SHVC with 2 and 3 layers for UHD deployment with HEVC and MPEG-4 AVC as base layers. The next chapter describes the need for rate allocation, SHVC rate allocation problem and its evaluation for 2 layers for a sample client distribution.

Chapter 6

SHVC RATE ALLOCATION

6.1 Need for Rate Allocation

Determining the effective allocation of bits into scalable layers is a tedious process, as shown in the Chapter 5. Also, this bitrate allocation into layers is dependent on the client distribution. The clients can support different resolutions (HD or UHD), codecs (HEVC or MPEG-4 AVC) and also have varying bandwidth characteristics. To effectively satisfy the bandwidth requirements of a given client distribution, an efficient rate allocation algorithm is necessary. This algorithm should determine the optimum number of layers and the optimal bitrate of each layer in the SHVC bit stream. For this purpose, a literature survey of rate allocation algorithms [38-48] for scalable video coding was done and an existing rate allocation problem was adapted to suit the deployment needs and was evaluated for a sample client bandwidth distribution.

6.2 Rate Allocation Problem

Hsu et al. formulated a bitrate allocation problem in the context of multiple layers or versions and a presented a solution based on dynamic programming [39]. This rate allocation problem is adapted for the scenario of multilayer streaming with SHVC by setting the number of layers to 2. The inputs and the outputs of this rate allocation is represented in Figure 6-1.

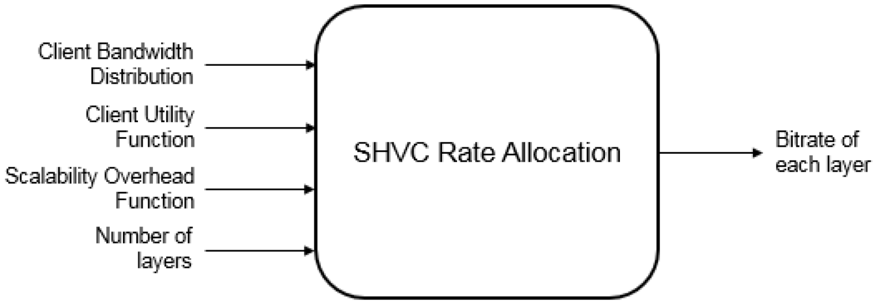


Figure 6-1 Inputs and Outputs of SHVC Rate Allocation

The scalability overhead function takes into account the overhead due to reduced coding efficiency and additional codec protocols. The client distribution represents the distribution of clients in the HTTP video streaming scenario having varying bandwidth characteristics. The client utility function represents the utility function of the given client distribution that best satisfies the client according to given utility such as PSNR or received bandwidth. The number of layers represents the number of layers in SHVC and the bitrate of each layer represents the optimal bitrate of each layer in SHVC to satisfy a given client distribution.

The problem formulation [39] is adapted for SHVC rate allocation. The total number of clients in the context of HTTP streaming can be divided onto C classes based on bandwidth and each of this class has probability mass distribution given by f(c). The SHVC bit stream can be structured into 2 layers. The goal is to find an optimum structuring policy $P^* = \{r_i, i=1, 2\}$, where r_i is the bitrate of each layer that yields maximum system wide utility U_o^* over the entire client class. This can be written as:

$$U_o^* = \max_P U_0 = \sum_{k=1}^{C} f(k) \cdot u(\overline{b_k}, b_k) \tag{6.1}$$

such that $r_1 < r_2$

The client utility function assumes that higher the effective rate that a client receives, the more satisfied that client will be, and it is given by:

$$u_{rate}(\overline{b_c}, b_c) = \overline{b_c} \tag{6.2}$$

In Equation 6.2, $\bar{b}_c$ is the effective rate of the received stream and b_c is the available bandwidth of the client class c.

For a scalable bit-stream, the effect of scalability overhead can be modeled using overhead function ‘a’ and the effective rate of stream. The function ‘a’ specifies the fraction of the total stream rate that does not contribute to the video playback quality. The scalability overhead function depends on the characteristics of the video sequence, granularity of the scalable coding such as Fine-Grain Scalability (FGS) and Coarse-Grain Scalability (CGS), and rate of the layer being encoded as well as the rates of the previous layers. The effective rate $\bar{r}$ of the scalable bit-stream is equal to the rate r of the non-scalable stream that produces the same quality [39].

For a scalable bit-stream having CGS layers, the effective rate of layer l (1 ≤ l ≤ L), is defined as:

$$\overline{r_l} = \begin{cases} r_1, & l = 1 \\ \overline{r_{l-1}} + \dfrac{(r_l - r_{l-1})}{1 + a(r_l)}, & 2 \le l \le L \end{cases} \qquad (6.3)$$

Also,

$$\overline{b_c} = \overline{r_l}, \text{ when layers are CGS} \qquad (6.4)$$

6.3 Evaluation of Rate Allocation for 2 layers of SHVC

The rate allocation problem in Section 6.2 is evaluated for a sample client distribution given in Table 6-1, for 2 layers of SHVC. The scalability overhead function 'a' for PeopleOnStreet test sequence with CGS layers is defined as:

$$a = \max\{0.05 - 0.0000016 * r_l, 0\} \qquad (6.5)$$

Table 6-1 Sample Client Distribution

Client Class	Bandwidth (Mbps)	Number of Clients	Probability Mass Function f(c)
C1	8	60	0.6
C2	12	10	0.1
C3	16	20	0.2
C4	20	10	0.1

This resulted in optimal base layer bitrate of r_1 = 8 Mbps and enhancement layer bitrate of r_2 = 16 Mbps. This evaluation of rate allocation for 2 layers works well for the sample client distribution.

6.4 Summary

This chapter describes the need for rate allocation algorithm for SHVC. An existing rate allocation problem is reviewed and is adapted to the scenario of SHVC with 2 layers. This rate allocation problem is evaluated for a sample client distribution for 2 layers of SHVC. The next chapter provides the conclusions of the research work and further work that can be done.

Chapter 7

CONCLUSIONS AND FURTHER WORK

The investigation of SHVC for UHD deployment in the context of HTTP streaming is performed by comparing it with Simulcast and Single Layer Coding of HEVC. The bitrate savings and scalability overhead for the deployment scenario of 2 and 3 layers of SHVC were determined. From these results, it can be concluded that UHD video can be efficiently deployed with scalable video coding SHVC.

Various experiments were conducted to observe the effect of varying ratio of bit allocation into base and enhancement layers. Previous investigations on Scalable Video Coding indicated that adding more bits into base layer reduces the scalability overhead [42]. However, from the experiments conducted, it is determined that this is not true, as SHVC scalability overhead depends on the ratio bit rate allocation into base and enhancement layers. Additionally, an existing bitrate allocation algorithm is adapted and evaluated in the context of 2 layered SHVC for a sample client distribution and optimal bitrates for base and enhancement layers are obtained. This research work is published as an IEEE conference paper [21].

Further, work on exploring optimal bitrate algorithms for allocation of bits into layers of SHVC based on Game theory and other approaches can be done, considering various scalability options (such as spatial, quality and combined scalabilities). Additional experiments to study the effect of scalability overhead for its modeling can be done using several test sequences. Also, evaluation of SHVC for its computational complexity can be done and parallel processing techniques for encoding base and enhancement layers in SHVC can be explored.

Appendix A

Steps for SHVC Encoding

A.1 Steps for Encoding and Decoding with SHM6.1 (SHVC)

Obtaining the source code and compiling:

- For obtaining SHM6.1 (SHVC Reference Software) - Checkout the source code from https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-6.1/
- For obtaining JM18.6 (MPEG-4 AVC Reference Software) - Checkout the source code from http://iphome.hhi.de/suehring/tml/download/
- Compile the source code on Windows/Linux platform

Sample configuration:

Base layer (BL) – 1080p, Bitrate = 4Mbps
Enhancement Layer (EL) – 4k, Bitrate = 4Mbps
Number of layers = 2

Steps for encoding SHVC with MPEG-4 AVC base layer

1. Using JM18.6, encode the 1080p resolution test sequence for a bit rate of 4Mbps

   Parameters to change in the configuration file:

   - RateControlEnable = 1
   - Bitrate = 4000000

   This encoding results in bit stream file (sample.264) and a reconstructed video (sample_rec.yuv).

2. Using SHM6.1, encode the 4K resolution test sequence for a bit rate of 4Mbps for EL. Use the reconstructed file (sample_rec.yuv) from step 1 for specifying -ibl parameter while encoding with SHVC.

   Sample command line for Encoding (Linux platform):

   ```
   ./TAppEncoderStatic /VideoCoding/05_SHM/cfg/encoder_randomaccess_main.cfg
   -b 01_results/ParkJoy_MPEG-4 AVC_HEVC_RA_Main_UHD_4_4mbps.bin
   -c /VideoCoding/05_SHM/cfg/per-sequence-svc-avcbase/ParkJoy-2x.cfg
   ```

```
-c /VideoCoding/05_SHM/cfg/layers_avcbase.cfg --TargetBitrate1=4000000 -ibl
/VideoCoding/AVC/bin/01_results/ParkJoy_HD_AVC_3B_4mbps_rec.yuv
>01_results/01_ParkJoy_AVC_HEVC_RAMain_UHD_4_4mbps.txt &
```

3. Use SHM6.1 decoder with the following command line (Linux platform):

```
./TAppDecoderStatic -b 01_results/ParkJoy_AVC_HEVC_RA_Main_UHD_4_4mbps.bin
-ls 2 -ibl /VideoCoding/AVC/bin/01_results/ParkJoy_HD_AVC_3B_4mbps_rec.yuv
-o1 01_results/ParkJoy_AVC_HEVC_RAMain_UHD_4_4mbps.yuv
```

Steps: SHVC with HEVC base layer

1. Use SHM6.1 Encoder with the following command line (Linux platform)-

```
./TAppEncoderStatic -c /VideoCoding/05_SHM/cfg/encoder_randomaccess_main.cfg
-b 01_results/ParkJoy_HEVC_HEVC_RA_Main_4_4mbps.bin
-c / VideoCoding/05_SHM/cfg/per-sequence-svc/ParkJoy-2x.cfg
-c /VideoCoding/05_SHM/cfg/layers.cfg
--TargetBitrate0=4000000 --TargetBitrate1=4000000
>01_results/01_results_ParkJoy_HEVC_HEVC_RAMain_HD_UHD_4_4mbps.txt &
```

2. Use SHM6.1 decoder with the following command line (Linux platform)-
3.

```
./TAppDecoderStatic -b 01_results/ParkJoy_HEVC_HEVC_RA_Main_4_4mbps.bin
-ls 2 -o0 01_results/ParkJoy_HEVC_HEVC_RAMain_HD_4_4mbps.yuv
-o1 01_results/ParkJoy_HEVC_HEVC_RAMain_UHD_4_4mbps.yuv
```

Appendix B

Source Code for BD-Metrics

SHVC investigation involves conducting numerous experiments and obtaining R-D plots. So the process of encoding, extracting and plotting results are automated using a combination of Bash and Python scripts. Bash scripts are written to automate the encoding process. Python scripts are written to extract the bitrate and PSNR values from the encoding output file and these results are tabulated in excel book. Also, python scripts are used for computation of BD-Metrics and plotting R-D curves.

- **Python Script for BD-Metrics computation - BDmetrics.py**

```
#!/usr/bin/python
# Bjontegaard's metric allows to compute the average gain in PSNR or the
# average per cent saving in bitrate between two rate-distortion curves
# Reference 1: http://www.mathworks.com/matlabcentral/fileexchange/41749-bjontegaard-metric-calculation--bd-psnr-/content//bjontegaard2.m
# Reference 2: http://vpx-codec comparison.webm.googlecode.com/git/visual_metrics.py
        import numpy
        import math
        def bd_psnr(rate1, psnr1, rate2, psnr2):
          log_rate1 = map(lambda x: math.log(x), rate1)
          log_rate2 = map(lambda x: math.log(x), rate2)
          # Cubic poly fit of RD points
          p1 = numpy.polyfit(log_rate1, psnr1, 3)
          p2 = numpy.polyfit(log_rate2, psnr2, 3)
          # Integration interval
          min_int = max([min(log_rate1),min(log_rate2)])
          max_int = min([max(log_rate1),max(log_rate2)])
          # Find integral
          p_int1 = numpy.polyint(p1)
```

```
  p_int2 = numpy.polyint(p2)

  int1 = numpy.polyval(p_int1, max_int) - numpy.polyval(p_int1, min_int)

  int2 = numpy.polyval(p_int2, max_int) - numpy.polyval(p_int2, min_int)

  # Find the average difference

  avg_diff = (int2 - int1) / (max_int - min_int)

  return avg_diff

def bd_bitrate(rate1, psnr1, rate2, psnr2):

  log_rate1 = map(lambda x: math.log(x), rate1)

  log_rate2 = map(lambda x: math.log(x), rate2)

  # Cubic poly fit of RD points

  p1 = numpy.polyfit(psnr1, log_rate1, 3)

  p2 = numpy.polyfit(psnr2, log_rate2, 3)

  # Integration interval

  min_int = max([min(psnr1),min(psnr2)])

  max_int = min([max(psnr1),max(psnr2)])

  # Find integral

  p_int1 = numpy.polyint(p1)

  p_int2 = numpy.polyint(p2)

  int1 = numpy.polyval(p_int1, max_int) - numpy.polyval(p_int1, min_int)

  int2 = numpy.polyval(p_int2, max_int) - numpy.polyval(p_int2, min_int)

  # Find the average difference

  avg_exp_diff = (int2 - int1) / (max_int - min_int)

  # Convert to a percentage.

  avg_diff = (math.exp(avg_exp_diff) - 1) * 100

  return avg_diff
```

Appendix C

Acronyms and Abbreviations

AVC – Advanced Video Coding

BL – Base Layer

CABAC – Context Adaptive Binary Arithmetic Coding

CDN – Content Delivery Network

CGS – Coarse Grain Scalability

CPB – Coded Picture Buffer

CPU – Central Processing Unit

CTB – Coding Tree Block

CTU – Coding Tree Unit

CU – Coding Unit

DASH – Dynamic Adaptive Streaming over HTTP

DCT – Discrete Cosine Transform

DPB – Decoded Picture Buffer

DST – Discrete Sine Transform

EL – Enhancement Layer

FIR – Finite Impulse Response

fps – Frames per second

HD – High Definition

HDS – HTTP Dynamic Streaming

HEVC – High Efficiency Video Coding

HLS – High Level Syntax (HEVC)

HLS – HTTP Live Streaming (Apple streaming)

HRD – Hypothetical Reference Decoder

HTTP – Hyper Text Transfer Protocol

IEC – International Electro-technical Commission

ILR – Inter Layer Reference

ISO – International Organization for Standardization

ITU-T – International Telecommunication Union - Telecommunication Standardization sector

JCTVC – Joint Collaborative Team on Video Coding

LUT – Look-Up Table

LTE – Long Term Evolution

Mbps – Megabits per second

MPD – Media Presentation Description

MPEG – Moving Picture Experts Group

MV – Motion Vector

NAL – Network Abstraction Layer

OTT – Over the Top

PDA – Personal Digital Assistant

POC – Picture Order Count

PPS – Picture Parameter Set

PSNR – Peak Signal to Noise Ratio

PU – Prediction Unit

QP – Quantization Parameter

RD – Rate Distortion

RL – Reference Layer

RTP – Real Time Protocol

RTSP – Real Time Streaming Protocol

SAD – Sum of Absolute Differences

SAO – Sample Adaptive Offset

SATD – Sum of Absolute Transformed Differences

SD – Standard Definition

SEI – Supplemental Enhancement Information

SHVC – Scalable High Efficiency Video Coding

SNR – Signal to Noise Ratio

SPIE – Society of Photo-Optical and Instrumentation Engineers

SPS – Sequence Parameter Set

SSD – Sum of Squared Differences

TMVP – Temporal Motion Vector Prediction

TU – Transform Unit

UHD – Ultra High Definition

URL – Uniform Resource Locator

VCEG – Video Coding Experts Group

VCL – Video Coding Layer

VPS – Video Parameter Set

WPP – Wave-front Parallel Processing

References

[1] G. J. Sullivan et al, “Overview of the High Efficiency Video Coding (HEVC) Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] J. Chen et al., “HEVC Scalable Extensions (SHVC) Draft Text 7”, Document JCTVC-R1008, Joint Collaborative Team on Video Coding (JCT-VC), 18th JCT-VC Meeting, Sapporo, 30 Jun. – 9 Jul. 2014.

[3] G. J. Sullivan et al, “Standardized extensions of High Efficiency Video Coding (HEVC)”, IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[4] J.M. Boyce et al, “Overview of SHVC: Scalable Extensions of the High Efficiency Video Coding (HEVC) Standard”, IEEE Transactions on Circuits and Systems for Video Technology, [Early Access], pp.1-1, 2015.

[5] JVT Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC), Mar. 2003, JVT-G050 [Online] Available: http://ip.hhi.de/imagecom_G1/assets/pdfs/JVT-G050.pdf

[6] V. Sze, M. Budagavi and G.J. Sullivan, “High Efficiency Video Coding (HEVC): Algorithms and Architectures”, Springer, 2014.

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

[8] M. Wien, “High Efficiency Video Coding: Coding Tools and Specification”, Springer, 2015.

[9] I.E.G. Richardson, “Video Codec Design: Developing Image and Video Compression Systems”, Wiley, 2002.

[10] I.E.G. Richardson, “H.264 and MPEG-4 Video Compression”, Hoboken, NJ, Wiley, 2003.

[11] V. Sze and M. Budagavi, “Design and Implementation of Next Generation Video Coding Systems (H.265/HEVC Tutorial)”, IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, June 2014. [Online] Available: http://www.rle.mit.edu/eems/publications/tutorials/

[12] F. Pescador et al, "Complexity analysis of an HEVC decoder based on a digital signal processor," IEEE Transactions on Consumer Electronics, vol. 59, no. 2, pp. 391-399, May 2013.

[13] Y-J. Ahn et al, “Implementation of fast HEVC encoder based on SIMD and data-level parallelism”, EURASIP Journal on Image and Video Processing 2014, Mar. 2014. [Online] Available: http://jivp.eurasipjournals.com/content/2014/1/16

[14] Tutorial on HEVC by S. Riabtsev, “Detailed Overview of HEVC/H.265”. [Online] Available: https://app.box.com/s/rxxxzr1a1lnh7709yvih

[15] J. Sole et al, "Transform Coefficient Coding in HEVC", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1765-1777, Dec. 2012.

[16] C. C. Chi et al, “Parallel scalability and efficiency of HEVC parallelization approaches,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1826-1837, Dec. 2012.

[17] F. Henry and S. Pateux, “Wavefront parallel processing”, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-E196, Mar. 2011.

[18] I. Sodagar, “The MPEG-DASH Standard for Multimedia Streaming over the Internet,” IEEE Multimedia, vol. 18, no. 4, pp. 62-67, Apr. 2011.

[19] J. Chen et al, “Scalable HEVC (SHVC) Test Model 6 (SHM 6)”, Document JCTVC-Q1007, Joint Collaborative Team on Video Coding (JCT-VC), 17th JCT-VC Meeting, Mar. /Apr. 2014.

[20] Y. Ye and P. Andrivon, “The Scalable Extensions of HEVC for Ultra-High-Definition Video Delivery”, IEEE Multimedia, vol. 21, no. 3, pp. 58-64, Jul.-Sept. 2014.

[21] U.S.M. Dayananda and V. Swaminathan, “Investigating Scalable High Efficiency Video Coding for HTTP streaming”, IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2015, pp. 1-6, Jun.-Jul. 2015.

[22] S.-F. Tsai et al, “A 1062 Mpixels 8192x4320p high efficiency video coding (H.265) encoder chip”, Symposium on VLSIC, pp. 188-189, 2013.

[23] R. Takada et al, “Improvement of 8K UHDTV Picture Quality for H.265/HEVC by Global Zoom Estimation”, IEEE International Conference on Consumer Electronics (ICCE 2015), Jan. 2015.

[24] H. Lee et al, “Scalable Extension of HEVC for Flexible High-Quality Digital Video Content Services", ETRI Journal, vol. 35, no. 6, pp. 990-1000, Dec. 2013.

[25] Tutorial on HEVC by D. Grois et al, “HEVC/H.265 Video Coding Standard including the Range Extensions Scalable Extensions and Multiview Extensions”, IEEE International Conference on Image Processing (ICIP), 2015. [Online] Available:https://datacloud.hhi.fraunhofer.de/owncloud/public.php?service=files&t=8edc97d26d46d4458a9c1a17964bf881 (Password: a2FazmgNK)

[26] H. Schwarz et al, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol.17, no.9, pp. 1103-1120, Sept. 2007.

[27] Y. Ye et al, "Manipulating Ultra-High Definition Video Traffic", IEEE Multimedia magazine, vol. 22, no. 3, pp. 73-81, July-Sept. 2015.

[28] Recommendation ITU-R BT.2020-1, “Parameter Values for UHDTV Systems for Production and International Programme Exchange”, Int'l Telecommunication Union, 2014.

[29] ST 2036-1, “Ultra High Definition Television - Image Parameter Values for Program Production”, SMPTE, 2013.

[30] Recommendation ITU-R BT.709, “Parameter Values for the HDTV Standards for Production and International Programme Exchange”, International Telecommunication Union, 2010.

[31] J. Chen et al, “Scalable Video Coding Extension for HEVC”, IEEE Data Compression Conference (DCC), pp. 191-200, Mar. 2013.

[32] P. Helle et al, “A Scalable Video Coding Extension of HEVC”, IEEE Data Compression Conference (DCC), pp. 201-210, Mar. 2013.

[33] M. Wien, H. Schwarz, and T. Oelbaum, “Performance analysis of SVC,” Special issue on Scalable Video Coding (SVC), IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1194-1203, Sept. 2007.

[34] I. Irondi, Q. Wang and C. Grecos, “Empirical evaluation of H.265/HEVC-based dynamic adaptive video streaming over HTTP (HEVC-DASH)”, Proceedings of SPIE, Real-Time Image and Video Processing, vol. 9139, May 2014.

[35] J. Nightingale, Q. Wang and C. Grecos, “Scalable HEVC (SHVC)-Based video stream adaptation in wireless networks”, IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), pp. 3573-3577, Sept. 2013.

[36] C. Muller et al, “Using Scalable Video Coding for Dynamic Adaptive Streaming over HTTP in mobile environments”, Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 2208-2212, Aug. 2012.

[37] W. Hamidouche et al, “4K real time video streaming with SHVC decoder and GPAC player”, IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1-2, July 2014.

[38] A. Unterweger and H. Thoma, “The influence of bit rate allocation to scalability layers on video quality in H.264 SVC”, Proceedings of IEEE Picture Coding Symposium (PCS 2007), Nov. 2007.

[39] C. H. Hsu and M. Hefeeda, “Optimal Coding of Multilayer and Multiversion Video Streams”, IEEE Transactions on Multimedia, vol. 10, no. 1, pp. 121-131, Jan. 2008.

[40] H. Lee et al, “Implementing rate allocation and control for real-time H.264/SVC encoding”, in Digest of Technical Papers, IEEE International Conference on Consumer Electronics (ICCE), pp. 269-270, Jan. 2010.

[41] J. Liu et al, “Bit Allocation for Spatial Scalability Coding of H.264/SVC With Dependent Rate-Distortion Analysis”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 7, pp. 967-981, July 2010.

[42] C. Hsu and M. Hefeeda, “Optimal partitioning of fine-grained scalable video streams”, Proceedings of ACM International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV’07), pp. 63–68, Jun. 2007.

[43] X. Wang et al, “Generalized Nash Bargaining Solution to Rate Control Optimization for Spatial Scalable Video Coding”, IEEE Transactions on Image Processing, vol. 23, no. 9, pp. 4010-4021, Sept. 2014.

[44] P. Baccichet et al, “Low-delay peer-to-peer streaming using scalable video coding”, Packet Video 2007, pp.173-181, Nov. 2007.

[45] I. Ahmad and J. Luo, “On using game theory to optimize the rate control in video coding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 2, pp. 209-219, Feb. 2006.

[46] Y. Chen, B. Wang and K. J. R. Liu, “A game-theoretic framework for multi-user multimedia rate allocation”, IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2009, pp.1997-2000, Apr. 2009.

[47] L. Li and H. Li, “λ Domain based optimal bit allocation for scalable high efficiency video coding," IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 2788-2791, May 2015.

[48] J. Liu et al, “Dynamic layering and bandwidth allocation for multisession video broadcasting with general utility functions”, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Society, INFOCOM, vol. 1, pp. 630-640, Mar./Apr. 2003.

[49] V. Seregin and Y. He, “Common SHM test conditions and software reference configurations”, Document JCTVC-Q1009, Joint Collaborative Team on Video Coding (JCT-VC), 17th JCT-VC Meeting, Mar./Apr. 2014.

[50] J. M. Nightingale et al, “A simulator tool set for evaluating HEVC/SHVC streaming”, SPIE Photonics West/EI, Paper 9400-22, San Francisco, CA, Feb. 2015.

[51] Special Issue on emerging research and standards on next generation video coding, IEEE Transactions on CSVT, vol. 22, pp. 1646-1909, Dec. 2012.

[52] G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves”, Technical Report VCEG-M33, ITU-T Video Coding Experts Group (VCEG), Mar. 2001.

[53] S. Pateux and J. Jung, “An Excel add-in for computing Bjontegaard metric and its evolution,” VCEG Meeting, Marrakech, Morocco, Jan. 2007.

[54] F. Bossen, “Excel template for BD-rate calculation based on Piece-wise Cubic Interpolation”, JCT-VC Reflector, 2011.

[55] BD Metrics computation using MATLAB source code [Online] Available: http://www.mathworks.com/matlabcentral/fileexchange/41749-bjontegaard-metric-calculation--bd-psnr-/content/bjontegaard2.m

[56] BD Metrics: VCEG-M34. [Online] Available: http://wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M34.xls

[57] SHVC Reference software – SHM6.1 [Online] Available: https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/

[58] HEVC Reference Software – HM 15.0 [Online] Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[59] HEVC Reference software manual [Online] Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.2-dev/doc/software-manual.pdf

[60] Test sequences [Online] Available: ftp://ftp.tnt.uni-hannover.de/testsequences/ and ftp://ftp.tnt.uni-hannover.de/scalable/sequences/

[61] Elecard HEVC Analyser [Online] Available: http://www.elecard.com/en/products/professional/analysis/hevc-analyzer.html

[62] JCT-VC documents [Online] Available: http://phenix.int-evry.fr/jct/

[63] HEVC reference [Online] Available: https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding

[64] Adaptive Streaming reference [Online] Available: https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming

Biographical Information

Uma received her Bachelor’s Degree in Electrical and Electronics Engineering from PES Institute of Technology, Bangalore, India in 2011. After her undergraduate studies, she worked as a Product Development Engineer for two years with Alstom Transport, Bangalore.

With a passion for Signal processing applications and Embedded Systems development, she pursued her Masters degree in Electrical Engineering from the University of Texas at Arlington from August 2013 to December 2015. During this period, she worked with Dr. K.R. Rao and was actively involved in research with the Multimedia Processing Lab. Her research interests include Video Compression and Streaming technologies.

She was awarded with Jack Fitzer Endowed Scholarship and Electrical Engineering Scholarship during her course of study at UT Arlington. She had an opportunity to work as Video Systems Research Intern at Systems Technology Lab, Adobe Research in San Jose. She plans to pursue her career in the field of Video and Embedded Systems development.