A COMPARATIVE ANALYSIS OF EMERGING VIDEO CODECS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OLGU KIVANÇ ONACAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

APRIL 2020

Approval of the thesis:

A COMPARATIVE ANALYSIS OF EMERGING VIDEO CODECS

submitted by **OLGU KIVANÇ ONACAK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Eng.** ________________

Prof. Dr. Gözde Bozdağı Akar
Supervisor, **Electrical and Electronics Eng., METU** ________________

Examining Committee Members:

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng., METU ________________

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Eng., METU ________________

Prof. Dr. Alptekin Temizel
Graduate School of Informatics, METU ________________

Assoc. Prof. Dr. Fatih Kamışlı
Electrical and Electronics Eng., METU ________________

Assoc. Prof. Dr. Hakkı Alparslan Ilgın
Electrical and Electronics Eng., Ankara Uni. ________________

Date: 22.04.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Olgu Kıvanç Onacak

Signature:

ABSTRACT

A COMPARATIVE ANALYSIS OF EMERGING VIDEO CODECS

Onacak, Olgu Kıvanç
Master of Science, Electrical and Electronics Engineering
Supervisor: Prof. Dr. Gözde Bozdağı Akar

April 2020, 189 pages

As screen technology and the internet evolve, video codecs should be improved to meet the requirements of video transfer. The use of AV1 codec, developed by AOM (Alliance for Open Media), which has recently come to the fore, is becoming widespread. Alternatively, ITU-T and ISO standard bodies are about to complete the development stages of the VVC (Versatile Video Coding) codec by making improvements to the HEVC (H.265) codec, which is already accepted by the codec community and widely used. Within the scope of this thesis, the configuration parameters of AV1, HEVC and VVC codecs are changed and comparisons are made both within themselves and against each other. Comparisons are made based on changing configuration parameters which are coding tree units (CTU), I-frame interval, and constant bit per second. For comparisons, 5 different characteristics CIF quality videos and a single 4K quality video are used. CIF quality videos are processed according to their temporal and spatial information, and the codecs are compared using these informations. PSNR, SSIM and BD-Rate measurement metrics are used while making comparisons.

Keywords: Video Codec, Video Processing, AV1, HEVC, VVC, Analysis

ÖZ

YENİ GELİŞTİRİLEN VİDEO KODEKLERİN KARŞILAŞTIRMALI ANALİZİ

Onacak, Olgu Kıvanç
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Danışmanı: Prof. Dr. Gözde Bozdağı Akar

Nisan 2020, 189 sayfa

Ekran teknolojisi ve internet dünyası geliştikçe videoların etkin bir şekilde transfer edilmesi, çevrim içi olarak izlenmesi ve paylaşılması için kullanılan video kodeklerin performanslarının arttırılması gerekmektedir. AOM (Alliance for Open Media) tarafından geliştirilen ve son zamanlarda ön plana çıkan açık kaynak kullanımına sahip AV1 kodeğinin kullanımı yaygınlaşmaktadır. Buna alternatif olarak ITU-T ve ISO firmaları hali hazırda kullanımı oldukça yaygın kabul gören HEVC (H.265) kodeğine iyileştirmeler yaparak VVC (Versatile Video Coding) kodeğinin geliştirme aşamalarını tamamlamak üzeridir. Bu tezin kapsamında AV1, HEVC ve VVC kodeklerinin konfigürasyon paramereleri değiştirilerek, hem kendi içlerinde hemde birbirlerine karşı karşılaştırmaları yapıldı. Karşılaştırmalar kodlama ağacı birimleri, I çerçeve aralığı ve sabit saniyedeki bit sayısı değişimine göre yapıldı. Karşılaştırmalar için 5 adet farklı karakterdeki CIF çözünürlükte videolar ve 1 adet 4K video kullanıldı. CIF videolar zamansal ve uzaysal değişimlerine göre 5 farklı kategoriye ayrıldı ve kodekler karşılaştırılırken bu değişimler üzerinden yorumlamalar yapıldı. Karşılaştırmalar yapılırken PSNR, SSIM ve BD-Rate ölçüm birimleri kullanıldı.

Anahtar Kelimeler: Video Kodek, Video İşleme, AV1, VVC, HEVC, Analiz

To my family

ACKNOWLEDGEMENTS

First, I would like to express my appreciation to my advisor, Prof. Gözde Bozdağı Akar, for the extreme support and guidance through my master's degree. Moreover, I would like to thank the remaining thesis examining committee Prof. İlkay Ulusoy, Prof. Alptekin Temizel, Assoc. Prof. Fatih Kamışlı and Assoc. Prof. Hakkı Alparslan Ilgın for their intelligent comments and encouragements.

I wish to thank my company Aselsan A.Ş. This work would never have been completed without my company's support and its graduate research policy.

I am deeply thankful to my parents, Sultan and Türkay Onacak, for their unconditional support and love through my entire education process from kindergarten to my master's degree. They are the kindest and most supportive parents that I would have ever hoped for.

Finally, I would like to thank to person who encourages me and had to put up with me through my thesis process, who continuously pushes me forward to be better in every aspect of my life; my beloved wife and my best friend, Ayça Onacak. There is no way that I could become who I am today without you.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

VVC	Versatile Video Coding
AOM	Alliance for Open Media
HEVC	High Efficiency Video Coding
PSNR	Peak Signal to Noise Ratio
GOP	Group of Pictures
CU	Coding Units
CTU	Coding Tree Unit
PU	Prediction Unit
TU	Transform Unit
SB	Super Block
SSIM	Structural Similarity Index

CHAPTER 1

INTRODUCTION

1.1. Motivation

The internet has become one of the most important aspect of the society, providing numerous different services that simplify the life of people. Two of the key features that internet provides are easy access and sharing of rich contents such as videos. Video transmission is currently taking %80 of the internet traffic and is growing day by day. Recently people have started to watch television not from the cable broadcasts but also from the internet broadcast providers such as NetFlix, Amazon etc. Besides, almost every application on our phones or tablets contain video transmission.

This growth in video transmission has caused increased data transmission rates and time in video transmission increased as a result. Download times also increased since the video resolutions are higher than before. For this reason, video transmission bitrates should be increased significantly in order to cope with that data increase [1]. The bandwidth of the internet providers cannot match with the proportion of the video size increase. Even if they can provide huge bandwidths such as 1000 Mbps, it is expensive that most individuals cannot afford it. Therefore, designing more efficient video codecs has become more valuable for engineers. Transferring the same video with less bandwidth and more efficiently is the key aspect of the video codec designers.

This research is motivated by a desire to investigate most recent developed and currently being improved video codecs AV1 and VVC and compare them with the cornerstone video codec HEVC (H.265) which everyone in the industry is familiar with.

1.1.1. Challenges for Future Video Codecs

The biggest challenge of the video codec systems is increasing the compression efficiency while sustaining the video quality and not losing any valuable information or not extending compression time. Most of the newly developed codecs tend to be more efficient in terms of compression efficiency. However, their encoding and decoding complexity gets higher. Therefore, the time to process encoding and decoding operations on the video increases drastically. If the video compression time is tried to be stabilized around some decent values, quality of video decreases when compression ratio increases.

Finding optimum values for compression ratio, bit rate compression time and video quality for on demand videos as well as real time videos is the main challenge that the codecs will face in the future.

1.2. Contributions

Principal contribution of this thesis is the design and implementation of a controlled test environment which is tailored specifically for the comparative analysis of video codecs HEVC, AV1 and VVC. The test bed contains various raw videos with quality and behavioral differences. These videos are given into codecs encoder and then decoded. The decoded videos are compared in terms of their PSNR and SSIM values. The overall results of encoding efficiency, bit rates, PSNR and SSIM values are evaluated and reported.

1.3. Thesis Outline

This chapter provides an overall understanding to the purpose of this thesis and the motivation behind it, states the codecs current status and future challenges as well as the contributions that the thesis explains.

Chapter 2 contains the background of the codecs and related works on the topic. It starts with describing how videos can be streamed in a more efficient way by designing better codecs and how these codecs can work together to be more applicable around the world. Afterwards, Chapter 2 goes into a general codec structure. It discusses every major element inside a codec. Prediction types are discussed first, then the types of filters that are used and the time when they are used are discussed. Entropy coding, quantization and transformation are explained finally. Overall, Chapter 2 provides a good understanding on a general codec and its features.

Chapter 3 is all about the methodology. Mainly, it explains the testbed design and the metrics used to carry on different tests on different videos. It also explains the video types and sizes as well as the platform specification that this thesis used for carrying the tests.

Chapter 4 discusses the results obtained from the tests. It starts with how these results are understood and interprets these results in detail. Moreover, it contains discussions about the results in detail.

Chapter 5 contains the overall conclusion and the future work that can be conducted as a continuation of this thesis.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1. Video Codecs

Videos are the largest internet consumption components. CISCO's global research data shows that video transfer occupies 59% of the total mobile internet traffic in 2017, and predicts that it will be around 79% by the end of 2022 [2].

In addition to real time streams, television platforms tend to lean on towards internet platforms such as NetFlix and Amazon, which supply on demand videos to the end user. Analog transmissions in televisions are decreasing every day, while the 21st century is becoming more internet dependent.

In the light of these information, researchers try to improve the efficiency of transferring, downloading and uploading videos. Video codecs are designed and used to fulfill this purpose, they transfer videos with high efficiency and low deprivation in the user experience.

There are multiple milestones in the history of video codecs. First ever codec like implementation was performed back in 1984 by Consultative Committee for International Telephony and Telegraphy (CCITT) which was H.120. H.120 and was never considered as a complete codec since it did not contain any motion estimation or compensation process. Moreover, it was never used commercially by companies. However, it was considered as a first step for evaluation of video codecs. Then International Communication Union (ITU) reorganized and merged CCITT into ITU-T which is the standardization section of ITU. After the merge, first big step was taken by ITU-T which is the evaluation of hybrid video compression standard H.261 which dates back to 1989 [3]. Features and operations were simple in H.261 but worked efficiently back in 1990s. After H.261 the video codec business started to ramp up and was taken seriously into consideration by engineers around the world. A couple of years later, another big company ISO emerged and released MPEG-1 standard. ISO and ITU-T standards became popular around the world as they are known as H.260 series and MPEG series standards. MPEG-2 was another milestone which was released more than 20 years ago. MPEG-2 provided good compression efficiency and was used so many years. After MPEG-2, MPEG-4 was released around 2000 and was accepted by the world and used as a good codec. H.264 was released around 2005. It was highly accepted and has been used immensely over 10 years. In 2015 ITU-T released another coding standard that is called H.265 (HEVC or HM) and still being used in today's world. Beside HEVC and H.265, Versatile Video Coding (VVC) was started to be developed in 2018 in order to be the future replacement for HEVC. VVC is currently still under development progress. Around 2000, Google also started to create video compression which is called VP codec. Since then Google developed codecs under the name of VP series and last one was VP9. Afterwards, a huge association which is called Alliance for Open Media (AOM) was formed with Google's initiative. This association gathered up big-name companies such as Google, NetFlix, Facebook and YouTube. In 2018, AOM started to develop AOM Video 1 (AV1) codec which is the successor for VP9. The overall timeline can be seen in Figure 2.1.

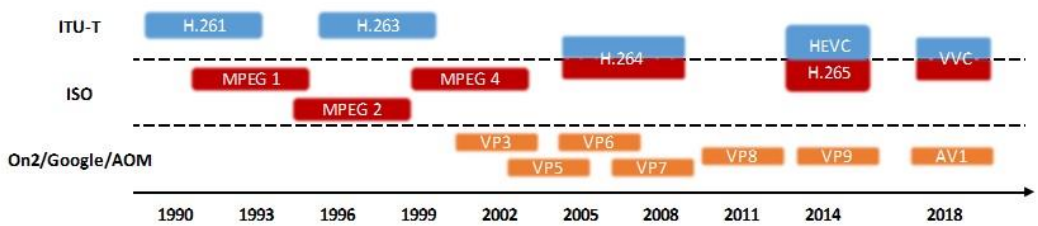

Figure 2.1. Timeline of video codecs [3]

2.2. Tools and Operations in Video Codecs

HEVC, AV1 and VVC codecs are all based on the generic Block-based Hybrid Coding Scheme [4]. Overall scheme can be seen in Figure 2.2. The input video is feed into group of pictures (GOPs), which consists of several picture frames. Each frame in GOP is then partitioned into smaller blocks, usually called as a Super Block (SB) or Coding Tree Block (CTB). Then the first picture in the GOP frame, called intra coded frame (I frame), is processed into transformation and quantization and then these frames are processed into entropy coding and fed into bit stream. Full information about I frame is processed into the bit stream in order to estimate the incoming frames from a based frame. This first frame is also stored in the memory of the encoder for further information and motion estimation. After I frame there are inter frames which are coded using inter predictions based on other inter frames or I frame. Then these inter coded frames are also processed through transformation, quantization and finally entropy coding stages. Further information about all the stages in a video codec are explained in the following sections.

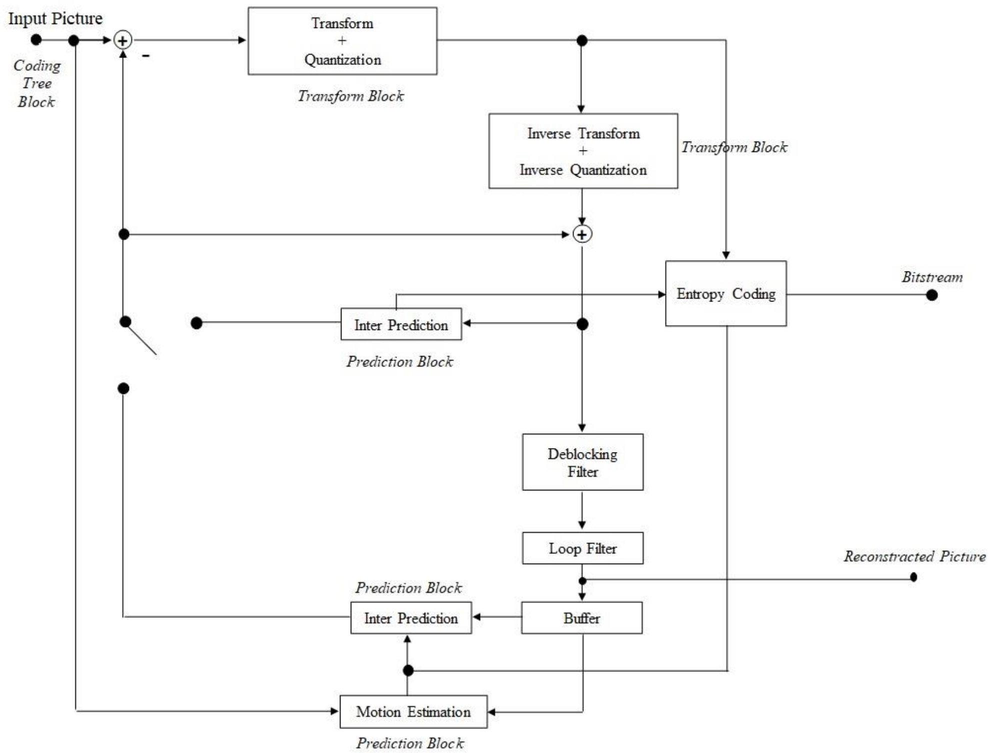

Figure 2.2. Block-based Hybrid Coding Scheme [4]

2.2.1. Block Partitioning

Video codecs perform block partitioning before any other steps. Block partitioning is a very essential process in order to divide the whole size of the frame into smaller parts. As expected, this portioning is performed following some rules for every single codec [5]. However, all perform the same tree structure in order to partition a frame. A tree structure means partitioning a frame into partitioned frames, then partitioning again and again until some logic is satisfied. All three codecs are different in terms of block partitioning.

2.2.1.1. HEVC Block Partitioning

HEVC introduced the flexible quad-tree partitioning of the coding tree units (CTUs) into coding units (CUs), prediction units (PUs) and transform units (TUs) of different sizes at different depths [4]. Figure 2.3 shows the partitioning form of the HEVC [6].

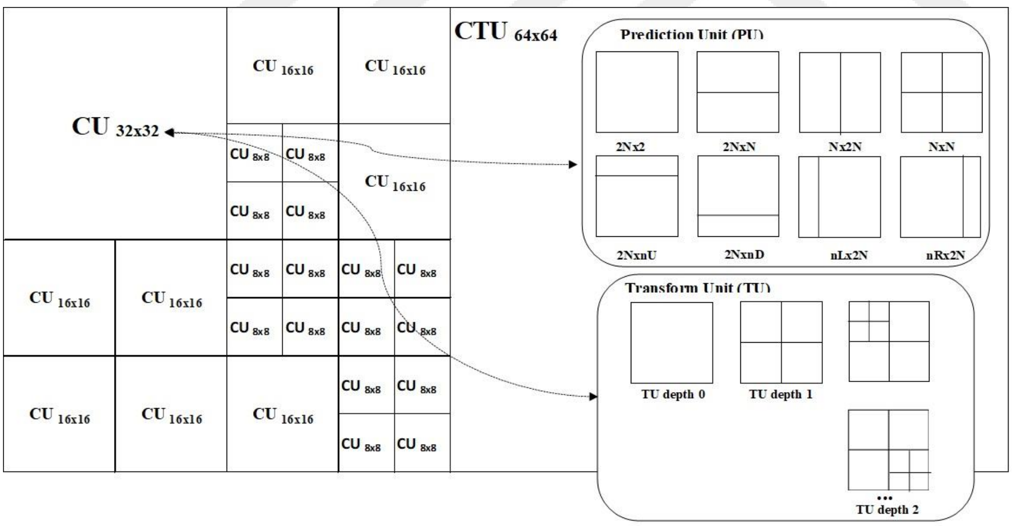

Figure 2.3. Coding tree units, prediction units and transform units of HEVC [6]

In HEVC standard coding tree units (CTUs) can be partitioned from 8x8 to 64x64. Before HEVC standard was introduced, the maximum CTU size was 16x16 [7]. Therefore, HEVC standard brings along way better coding efficiency when it was introduced. Once the CTU is formed, coding units (CUs) and coding blocks (CBs) are constructed from the CTU. These CU and CB can be further divided into smaller partitions.

Prediction units (PUs) are formed from CU. PU has its roots at CU level. The decision whether a block is PU or not is made by its root CU. These prediction units are combined to perform intra and inter predictions in the encoder [8].

Transform units (TUs) are constructed in a very similar way like a PU, TU has its roots at CU level. TU consists of the transformation and quantization coefficients. These coefficients are transformed by these TUs [7]. PU and TU size can be 4x4 minimum and 32x32 maximum depending on the partitioning and the root CTU size.

2.2.1.2. AV1 Block Partitioning

AV1 adopts a more sophisticated way to partition block than HEVC. In addition to quad-tree partitioning, AV1 uses extra binary-tree partitioning. The implementation is called quad-tree plus binary-tree (QTBT). In QTBT the root CTU is divided recursively using quaternary split into 4 equal parts, after this recursive partitioning binary partitioning is implemented to a square CTU and the leaf nodes are acquired [9]. The important point in AV1 block partitioning is that after binary partitioning is performed the partitioning process is stopped and leaf nodes are created.

The maximum size of the CTU is increased to 128x128 when it is compared with HEVC maximum size which is 64x64. Also, AV1 introduced 10 ways of expansion of the superblocks into smaller blocks which are shown in Figure 2.4. Prediction and transform units are not used in AV1 block partitioning.

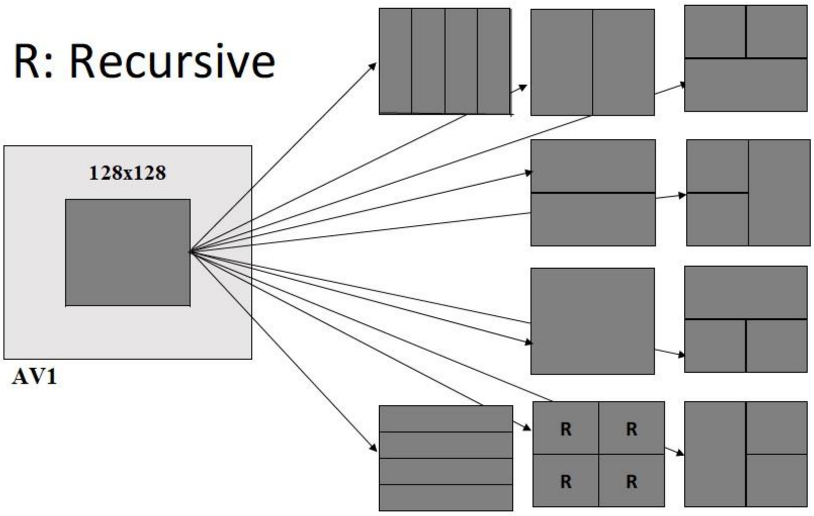

Figure 2.4. Partition tree in AV1 [9]

2.2.1.3. VVC Block Partitioning

VVC adopts a very similar block partitioning method to the HEVC's. Maximum CTU size is 128x128. In addition to quad tree partitioning in HEVC, VVC uses binary tree and ternary tree approaches as well, which is shown at Figure 2.5 [10]. Moreover, CTU partitioning in VVC is performed separately for luma and chroma samples in the same CTU, meaning that chroma samples will likely be partitioned less than luma samples. This increases the efficiency of the encoder. Minimum CU size for luma samples in VVC is 4x4 [10]. The splitting process is performed with two main steps, the first step is splitting the root CTU into smaller CTUs by using quaternary tree, as known as splitting into four equal parts. Each part can be recursively split into more equal 4 parts. When a CTU finishes quaternary partition, it either starts using binary and ternary partition or does not split at all. This further multi-type tree partitioning can be done recursively as well. Figure 2.6 shows an illustration of VVC block partitioning.

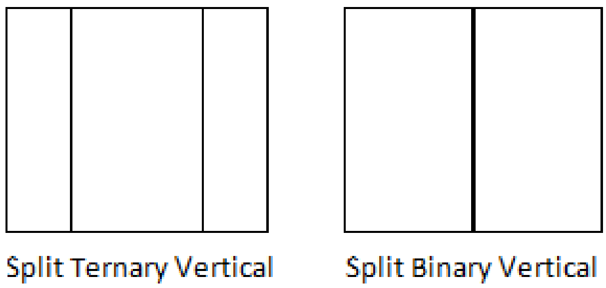

Figure 2.5. Difference between binary and ternary split [10]

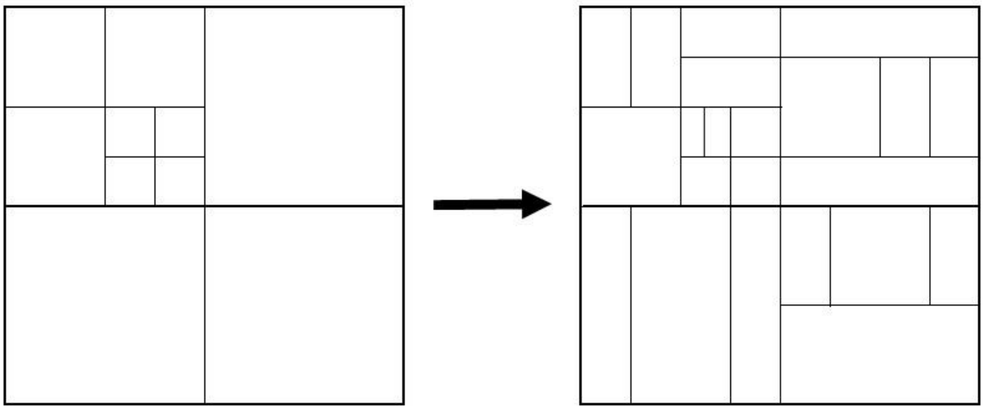

Figure 2.6. VVC partitioning example [10]

2.2.2. Quantization and Transformation

Quantization and transformations are performed in order to transfer the pictures into coding environment and the inverse quantization and transformations are used to display the resulting frames from the coding environment. Quantization and transformation processes are first processes to perform in the encoder in order to transfer the input signal to the discrete values in order to perform motion estimation and entropy coding.

2.2.2.1. HEVC Quantization and Transformation

HEVC uses very simple and few methods in transformation. Almost all blocks are transformed using Discrete Cosine Transform (DCT) beside 4x4 transform units (TUs) which is transformed by Discrete Sine Transform (DST) [11].

Both DCT and DST come from sinusoidal unitary transform family. They are considerably used to compress data in video processing field [12]. Moreover, DCT and DST have similar mathematical expressions; however, one important difference is that DCT starts with the first element cos(0) is equal to one, while DST needs a coefficient for its first element since sin(0) is equal to zero. Both DCT and DST are well known to discard small amplitude high frequency components which can cause artifacts in the decoded video. There are 8 versions of both DST and DCT. The difference between these versions are including coefficients or different cosine or sine inputs. Best known version of the DCT is the DCT-II which is the original version developed by Nasir Ahmed in 1972 [13]:

$$f_{DCT}(k) = a(k)\sum_{n=0}^{N-1} f(n)\cos\left(\frac{(2n+1)*\pi*k}{2*N}\right)$$

for $k = 0,1,2,3,\ldots,N-1$. $f(n)$ is the input signal and $f_{DCT}(k)$ is the resulting DCT output of the input signal. $a(k)$ is defined as:

$$a(k) = \begin{Bmatrix} \sqrt{\frac{1}{N}}\ for\ k = 0 \\ \sqrt{\frac{2}{N}}\ for\ k \neq 0 \end{Bmatrix}$$

2.2.2.2. AV1 Quantization and Transformation

In AV1 more sophisticated ways of transformations are used. AV1 uses 16 different vertical and horizontal combinations of DCT, Asymmetric DST (ADST), flipped ADST and Identity Transform (IDTX) [14].

Identity transform is the transformation type that copies entire data without changing any information, in AV1 identity transform is essential for sharp edges since not quantizing the data better visual experience can be acquired. Asymmetric ADST vanishes to zero at the prediction boundary, while becoming maximum at the end of boundary and this behavior is called asymmetric. Flipped ADST is the process that implements the reverse order of ADST, which is therefore maximum at the prediction boundary and minimum at the far end.

2.2.2.3. VVC Quantization and Transformation

VVC uses a different approach than both HEVC and AV1. It uses multiple different types of DCT and DST. Moreover, VVC performs Adaptive Multiple Transform (AMT).

AMT is implemented first in VVC as a primary algorithm for transformation on top of DCT-II or DST-II transformations. It is basically implementation of not only one or two DCT and DST transformation but all scale of transformations from DCT and DST family including DCT-II, DST-VII, DCT-VIII, DST-I and DCT-V [15]. The transformation selection is based on intra prediction mode selection for different type of coding units.

2.2.3. Intra Prediction

Intra prediction is the method to reduce spatial redundancy and to transform information within each block separately in order to increase coding efficiency. It is applied on the neighboring pixels inside the same partitioned block to reduce the energy before transform coding [16]. I frames are the intra coded frames, which consist of intra predictions only without using any other frames.

2.2.3.1. HEVC Intra Prediction

HEVC standard uses 33 different directional modes for intra prediction. In addition to directional modes HEVC uses both planar and flat (DC) modes [8].

Directional mode intra prediction was introduced to the video signaling industry very long time ago. It was first started as a prediction method between macroblocks in H.264 and former versions. With the HEVC the directional mode is implemented between coding units in the image. Moreover, the direction numbers increased with every new codec. In Figure 2.7 the angular modes can be seen for HEVC. First the coding unit check all the angular modes and try to find the best candidate for predicate itself on that unit. Best candidate is selected by calculating minimum Hadamard Transform Absolute Difference (HAD) cost [17]. It also means that the coding unit tries to find the best similar unit to reduce the information carried on itself. AV1 and VVC also adapts the directional mode intra prediction with more directions for better efficiency.

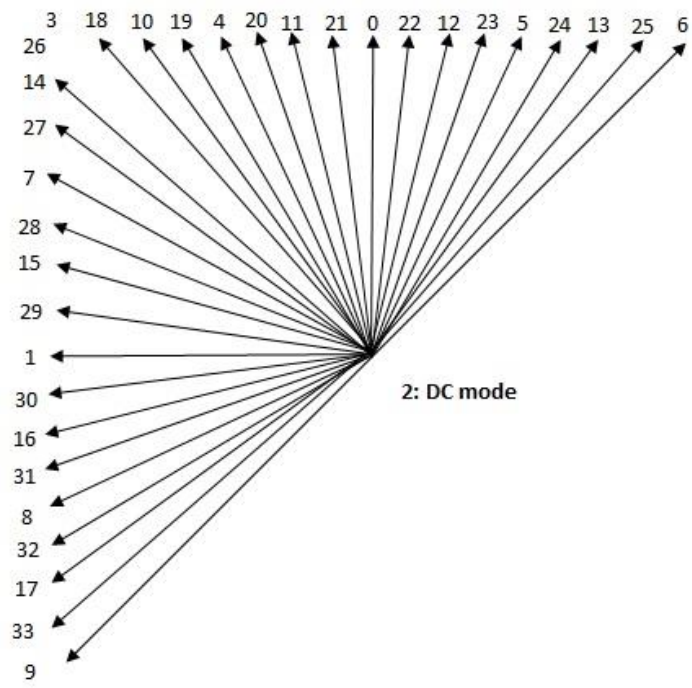

Figure 2.7. Angular directions of HEVC intra prediction [8]

2.2.3.2. AV1 Intra Prediction

AV1 applies the general angular mode prediction which is modified by considering of gradients and correlations inside the block and uses 56 directional modes. In addition to general angular mode, smooth predictor, Paeth predictor and chroma predicted from luma methods are used for intra prediction in AV1.

Paeth prediction is a simple prediction method that calculate a linear combination of three neighboring cells, which are left, top left and top to select the best cell as a predictor for upcoming operations.

Smooth predictor on the other hand calculates the weighted average of top and left blocks and calculates the interpolated pixels from bottom and right. Finally, it decides its successor for next operations [18].

Chroma predicted from luma method is a chroma only intra predictor, which predicts the chroma samples from reconstructed luma samples.

2.2.3.3. VVC Intra Prediction

VVC increase the directional modes even more than the AV1 by 67 directional modes including planar and DC modes. Furthermore, VVC uses Cross-Component Linear Model (CCLM).

CCLM is a new concept developed for VVC which calculates luma and chroma components within the block using a linear model [10]. Equation below shows how CCLM works.

$$P_c\,(i,j) = \; aR_l(i,j) + b$$

$P_c\,(i,j)$ is the predicted chroma pixel at the point (i,j), $R_l(i,j)$ is the reconstructed luma samples of the same location. Finally, a, b are the CCLM constants that derived by minimizing the regression error between the neighboring reconstructed luma and chroma samples around the current block.

2.2.4. Inter Prediction (Motion Compensation)

Inter prediction is the method to estimate the relation between frames especially using motion estimation and compensation methods to reduce spatial redundancy between frames [19]. Inter prediction is one of the most complicated and time-consuming operation of the video encoding. There are numerous different operations for inter prediction which codecs adapt.

2.2.4.1. HEVC Inter Prediction

HEVC uses hierarchical-weighted prediction using P and B frames, merge mode, block motion compensation and translation methods [20].

Hierarchical-weighted prediction is the method that assigns weights to the frames that will be used in the motion estimation for the current B or P frame, as well as assigning the order of the frames that are used in the process. Different hierarchical methods can be applied according to the I frame interval and GOP size.

Block motion compensation is the process that predicts a block from a block of equal size in the reference frame. After the block is predicted using prediction methods, motion prediction vector is created, and this prediction information is transferred to the decoder side rather than the blocks actual information.

HEVC encoder also enables PUs to share the motion information of neighboring PUs with flag to indicate whether using merge mode or not and index of merge. Up to 5 neighboring PUs can be used in the merge mode and they share the information of the motion prediction.

2.2.4.2. AV1 Inter Prediction

AV1 uses different approaches than HEVC and VVC. It uses advanced compound prediction model, warped motion compensation, dynamic spatial and temporal motion vector referencing and overlapped block motion compensation (OBMC) [21].

AV1 introduces new collection of compound prediction models to make the encoder more efficient and useful.

$$p(i,j) = p_1(i,j)k(i,j) - p_2(i,j)(1 - k(i,j))$$

The equation above illustrates how weighted prediction model works. $p(i,j)$ is the final joint prediction, $p_1(i,j)$ and $p_2(i,j)$ are two predictors, finally $k(i,j)$ is the weight associated for $p_1(i,j)$. Combination of these predictors are form the final predictor. This compound method is well known and used for all predictors; however, AV1 introduces four new concepts which are compound wedge prediction, difference-modulated masked prediction, frame distance-based compound prediction and compound inter-intra prediction.

- Compound wedge prediction method is predefining a codebook of 16 possible wedge partitions and to signal the wedge index in the bitstream when a coding unit chooses to be further partitioned.
- Difference modulated masked prediction method modulates the pixel difference between p1 and p2 with an absolute difference.

  $$p(i,j) = b + a|p_1(i,j) - \ p_2(i,j)|$$

  In the equation above, b controls how strongly one predictor is weighted over the other within the differing regions and a scaling factor, while a ensures a smooth modulation.
- Except non uniform weighting for predictors AV1 also utilizes uniform weight prediction with the method frame distance-based compound prediction. The weights are calculated as a function which takes distance as the main input. When the distance is further than the predicted frame, weight of that predictor decreases.
- Compound inter-intra prediction modes, which combine intra prediction p1 and single reference inter prediction p2, are developed to handle areas with emerging new content and old objects mixed.

Warped motion compensation is the method that combines local and global warped motion compensation. Global warped motion consists of camera motion estimation, while local warped motion aims to describe varying local motion.

In dynamic spatial and temporal motion vector referencing AV1 incorporates a sophisticated MV reference selection scheme to obtain good MV references for a given block by searching both spatial and temporal candidates. The motion field estimation process works in three stages: motion vector buffering, motion trajectory creation, and motion vector projection. Once all spatial and temporal candidates have been aggregated in the pool, they are sorted, merged and ranked to obtain up to four final candidates. To code a motion vector, AV1 uses a reference motion vector from the list, followed by the additional information to acquire the desired motion vector.

OBMC can largely decrease prediction errors near block edges by smoothly combining predictions created from adjacent motion vectors. In AV1, a 2-sided causal overlapping algorithm is designed to make OBMC easily fit in the advanced partitioning framework [9].

2.2.4.3. VVC Inter Prediction

VVC is very similar to HEVC in terms of inter prediction methods. VVC also uses hierarchical-weighted prediction using P and B frames, sub-coding unit (CU) based motion vector prediction (like HEVC PU-based motion vector prediction). In addition to those VVC uses affine motion prediction and decoder side motion vector refinement methods.

Conventional motion compensation using one motion vector that can only represent two-dimensional planar motion. This means that any block can be moved on the image plane in x and y directions only. However, in a natural video, strictly planar motion is quite rare, and things tend to move more freely (e.g. rotate and scale). VVC implements an affine motion model that uses two or three motion vectors to enable motion with four or six degrees of freedom for a block.

Another method that allows for the motion vectors to automatically be refined at the decoder without the transmission of additional motion data is to perform an actual motion search at the decoder side. While this basic idea has been around for a while, the complexity of a search at the decoder side was always considered too high until now. The process works in three steps. First, a normal bi-prediction is performed, and the two prediction signals are weighted into a preliminary prediction block. Using this preliminary block, a search around the position of the original block in each reference frame is performed. However, this is not a full search as an encoder would perform it, but a very limited search with a fixed number of positions. If a better position is found, the original motion vector is updated accordingly. Lastly, bi-prediction with the updated motion vectors is performed again to obtain the final prediction [10].

2.2.5. Filters

Deblocking and in loop filters are used by all three codecs. Deblocking filters are used to smoothen sharp edges on the frames and decrease the boundary artifacts between frames. In loop filters are generally used to filter undesired artifacts on the decoded frames in order to increase the visual quality of the reconstructed frames.

2.2.5.1. HEVC Filtering

HEVC uses simple deblocking filter and Sample Adaptive Offset Filter (SAO) together to filter undesired artifacts.

The most common type of filter that is being used in video codecs is deblocking filters. Deblocking filter operates rather basically. There is a boundary between signals for the filter and the filter checks the activity of both sides of the boundary; if the signal differences are very low at both sides of the boundary the filtering occurs and artifact is found in the system [22]. As we can see from the Figure 2.8, the q signals and the *p* signals are consistent, however the transition between *p* to *q* is above threshold and either *p* or *q* signals have boundary artifacts.

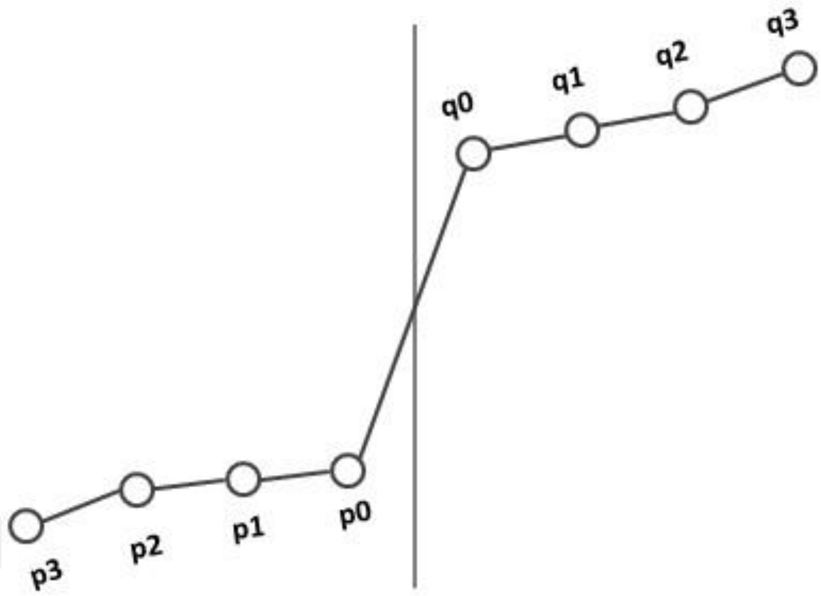

Figure 2.8. Block boundary for deblocking filter [22]

SAO filter, which is usually located after deblocking filters, operates for reducing the mean sample distortion of a block by classifying regions and calculating offsets for each region.

2.2.5.2. VVC Filtering

VVC is again very similar to HEVC; however, in addition to deblocking filter and SAO, VVC implements a more sophisticated adaptive loop filter (ALF). ALF consists of derivations of Wiener filters in order to minimize the artifacts after deblocking filters and SAO process the information.

2.2.5.3. AV1 Filtering

In addition to similar operations in VVC, AV1 performs more detailed and sophisticated methods which are film grain synthesis and frame super resolution [9].

Film grain synthesis is a method that acquires grains before the encoding process of the video is concluded, transfers the grains as a separate information. Decoder side obtains these grains and adds them into the original video at the decoding side. Film grain synthesis is a process that separates AV1 from its competitors.

Super resolution is a technique that helps to improve the quality of an image by applying upscaling techniques. Super resolution helps reducing aliasing and improves resolution as well as revive colors.

2.2.6. Entropy Coding

Entropy coding is used to compress the data without losing any information in order to transfer the code to the transfer environment. It is the last step before transferring out of the encoder and first step before decoding the incoming video in the decoder side. Entropy coders main purpose is to assign digitalized signals with common patterns to more efficient representors. Every entropy coder has its own rules and regulations to perform. However, the important thing is not losing a single information on the data.

2.2.6.1. HEVC Entropy Coding

HEVC uses context-adaptive binary arithmetic coding (CABAC) which is globally accepted and largely used. CABAC focuses on efficiently assigning unique and low-cost bit maps to the digitalized encoded video to save even more compression rate. It has mainly three components, which are binarization, context modeling and arithmetic coding [23]. Binarization is the process that assign binary representations to syntax elements. Context modeling is the estimation of these binaries to achieve even better compression with assigning most used binary clusters into simpler binary representation. Finally, arithmetic coding transforms these binaries into bits to be transferred.

2.2.6.2. AV1 Entropy Coding

AV1 has a different approach than CABAC. It uses multiplier-based context adaptive entropy coding and level map coefficient coding.

Multiplier-based context adaptive entropy coding uses a symbol-to-symbol adaptive multi-symbol arithmetic coder. Each syntax element in AV1 is a member of a specific alphabet of N elements, and a context that consists of a set of N possibilities together with a count to simplify fast early adaptation. AV1 changes to a level map design for sizeable transform coefficient modelling and compression [9].

2.2.6.3. VVC Entropy Coding

VVC adapts the CABAC and improves it with using context modelling for transform coefficient levels. The decoding decision in VVC uses a new two state model for each context instead of a look-up table (LUT) as in HEVC.

CHAPTER 3

EVALUATION AND TEST METHODS

3.1. Experiment Structure

The primary purpose of the experiments is to setup an environment that can be created by everyone. Therefore, experiments are controlled and reproductible in any time. Experiments give a better understanding of the video codecs by providing outcomes of different variable changes under standard codec settings for three codecs. To that end, I have compiled the reference codecs and perform codec operations on different type and resolution videos and the results of the videos have been compared in terms of the comparison metrics that were used in this thesis.

3.2. Testbed Design

In this part the decisions of the testbed design are explained. This includes the type of platform that is used, the types and characteristics of videos that are selected for the testing and the system that is used to perform encoding and decoding operations.

3.2.1. Platform Selection

Initially I have selected to perform operations on Ubuntu V18.04 LTS in a virtual box virtual machine. After several attempts and working on codecs there were problems occurred. The setup itself took so long to configure for a stranger into Linux environments, the steps were complicated, configuring the codecs were complicated, and there were plenty of dependencies to different software products. Moreover, the biggest disadvantage was to perform the encoding and decoding operations on a virtual machine. Even with utilizing full CPU power the operations took long time, with the virtualization the time for encoding and decoding unbearable.

After some search and looking into codecs websites, I realized that all three codecs support Windows Environment. All three codecs require Visual Studio 2015 or later versions. I compiled all three codecs with Visual Studio 2019.

I used latest stabilized versions of all three codecs in early 2020. All three codecs have been downloaded using their reference software pages. The version for VVC reference software is “VTM_VERSION 7.3”. HEVC version is “HM 16.21”. Finally, for AV1 reference software version is “AV1 1.0.0-errata1-avif-259-gfc038664e”.

3.2.2. System Specifications

I have used my personal hardware within the context of this thesis. System consist of four cores Intel i7 4710HQ with 2,50 GHz of single core speed, as well as 16 GB RAM. The important thing in the thesis content was to utilize CPU power for all three codecs equally in order to be able to measure and compare encoding times of the encoders. When comparing single encoder with different encoding parameter settings the system specification became subordinate since same encoder runs in the same system.

3.2.3. Video Selection

Video selection is the crucial part of the experiments since videos are very characteristic and unique in their own. One important aspect is to choose YUV video sequence. YUV is a color spectrum similar like RGB, however YUV has more human perception in it because it carries luminance (glow) component in it as well, therefore it eliminates compression artifacts and transmission errors better than RGB. Therefore, it reduces bandwidth of the transmission [24]. With all these benefits of YUV, usually codecs are performed with YUV video sequences.

In this thesis scope, I use YUV sequences in CIF (352x288) format. Arizona State University provides very well-known and cornerstone YUV sequences (such as Foreman) as an open resource in their video trace library [25]. These video sequences consist of all different movement and stationary behaviors and are very suited for codec comparison. For the sake of understanding the effects of parameters on different type of video sequences, the videos are classified. This classification is performed by calculating the temporal and spatial information of videos. Five different characteristics of videos were selected. In addition to these five CIF format videos, single 4K video has been used and its results have been compared with CIF video results. This 4K that consists of both local and camera movements was selected as famous “basketball” sequence. The CIF videos were used to compare the results of the codecs on different behavior of videos, where 4K video was used to compare codecs with each other as well as to compare codecs performances for lower and higher resolution videos.

3.2.3.1. Temporal and Spatial Information

Specification metrics temporal and spatial information are calculated within the video in order to classify videos into groups that have similar behavior. Temporal behavior is the difference between frames in the video at the time domain. More specifically bigger the temporal difference results in more movement behavior and mobility in the video.

$$M_n(i,j) = F_n(i,j) - F_{n-1}(i,j)$$

$$TI_n = std(M_n)$$

$$TI = mean(TI_n)$$

In above equations F_n is the nth frame in the video while i, j are the pixel locations. M_n is the difference between two consecutive frames and temporal information of nth frame is calculated taking standard deviation of the M_n for all pixels. Total TI is calculated by taking the mean for every frame in the video sequence. Figure 3.1 illustrates the importance of temporal information. First sequence “Bus” has both local and camera movement and the movement can be easily detected, while with the same frame movement there is not any significant change in the video “Container”.

Therefore, with the increased temporal information the codecs tend to increase their bitrate in order to maintain good quality for the videos.

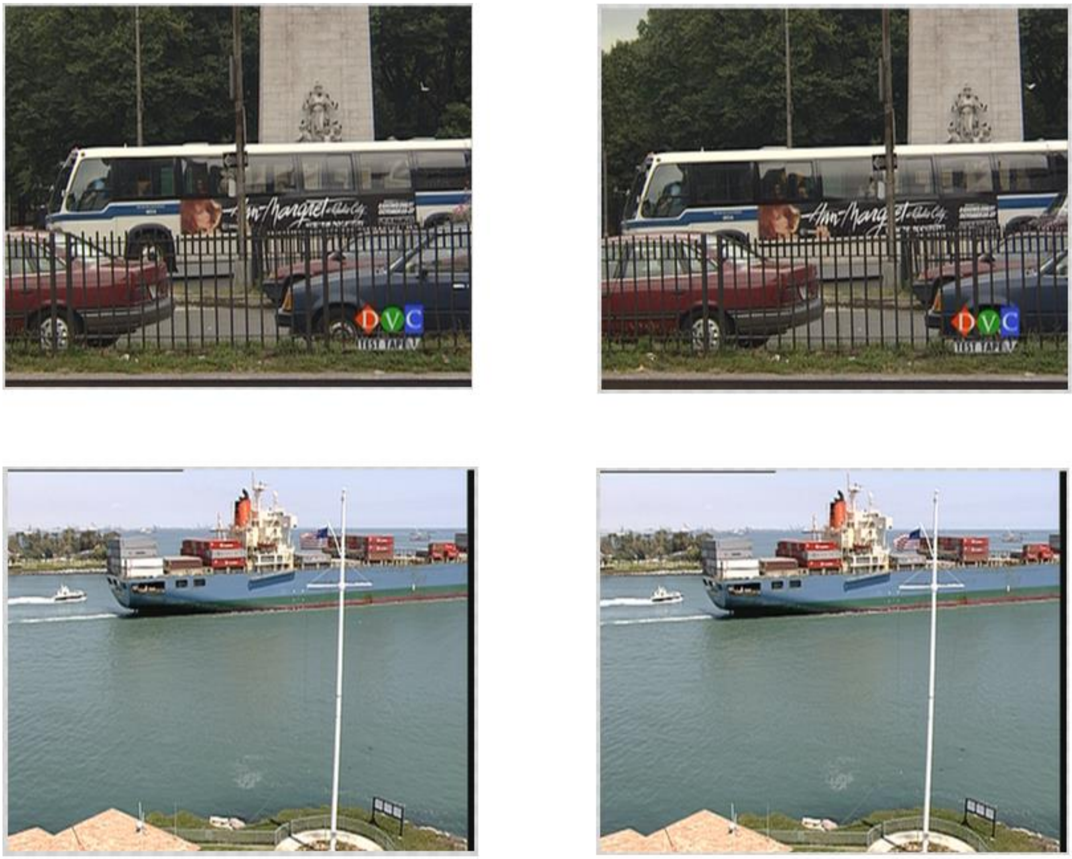

Figure 3.1. 10 frame difference between video sequences of “Bus” and “Container”

Spatial information measures the edge movement in the spatial domain. Specifically, it measures the edge deficiencies, distortion and blurriness at the edges of the frames. Spatial information is related with the complexity of the video in the frequency domain, greater spatial information results in greater complexity [26].

$$SI_n = std(Sobel(F_N))$$

$$SI = mean(SI_n)$$

SI is calculated by above equations, where SI_n represents SI value for specific frame and calculated such that by applying Sobel filter to every pixel in the frame and taking the standard deviation in the frame. After that total SI is calculated by taking the mean for every frames SI value.

3.3. Experimental Metrics

The experiments drawing results by using several different metrics, which are peak signal-to-noise ratio (PSNR) value, structural similarity index (SSIM), bit rates and encoding and decoding times. These metrics are usually incorporated together to decide whether an algorithm or decoder is better than the other. One metric alone can be deceptive since all metrics are dependent each other. Providing way better PSNR values without increasing encoding time is nearly impossible since possibly its complexity in intra and inter level prediction gets more complicated than another; therefore, taking into consideration all three metrics together is the crucial part of determining ramifications.

3.3.1. Peak Signal to Noise Ratio (PSNR)

There are two ways of determining how much the decoded video is like the original one. First one is objective video quality measurements. Objective video quality measurements are adhered to mathematical calculations of each frame between the original video and the decoded video. These mathematical calculations are differing from one measurement type to another. The most known and used method is PSNR. PSNR is widely used because of its simplicity. However, PSNR value is considered to be very disadvantageous, because it undervalues human perceptual visual system characteristic and the viewing conditions such as viewing angle and distance [27]. Moreover, PSNR can be computed differently for luma and chroma components. On the other hand, PSNR value is consistent over various types of videos, and it gives a better understanding to the reader, since it is widely used.

There are plenty other objective quality measurements such as SSIM and video quality metric (VQM). SSIM gives more accurate results than PSNR when compared with human perspective; however, SSIM does not sustain consistency. Sometimes SSIM gives such great results to the bad video encodings making SSIM not a trustworthy measurement metric [28]. On the other hand, VQM is the nearest measurement system to the subjective video quality analysis, since the main aspect of VQM is to track subjective video quality by using optimized objective to subjective correlation and relation [29]. PSNR and SSIM are used for video quality measurement technic within the scope of this thesis experiments.

PSNR calculation is based on the difference between original signal and noised signal. This difference is mean squared throughout every image and found and mean square error (MSE) which can be calculated as follows:

$$MSE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(S(i,j) - S_d(i,j))^2}{MN}$$

On the equation $S(i,j)$ refers to original signal and $S_d(i,j)$ refers to decoded (noise) signal, M and N defining the dimensions of the frame meaning width and height [30].

$$PSNR_k = 10\log\frac{(2^b - 1)^2}{MSE}$$

After finding MSE, calculation of PSNR is straight forward. In above equation b refers to bit depth, and k refers to each frame. For all frames logarithmic base value calculated and taking average result into overall PSNR of the video sequence. Higher PSNR values result better reconstructed video sequences since the difference between each pixel of original and reconstructed gets smaller.

PSNR calculation is accepted and used by majority of the video coding companies, such that they include automatic calculation of the PSNR between the encoded video and original video.

3.3.2. Structural Similarity Index (SSIM)

SSIM is more subjective observation between videos than PSNR calculation. SSIM is designed to improve traditional methods. SSIM calculates the similarity index between two images, SSIM considers that one image is perfect and calculates the other one from a fraction of the perfect image. SSIM calculates three main components which are luminance, contrast and structure.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{{\mu_x}^2 + {\mu_y}^2 + C_2}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

$$SSIM(x,y) = l(x,y)c(x,y)s(x,y)$$

Above equations μ_x, μ_y are average values that are average pixel values of 2 equal size NxN windows which are x, y. σ_x, σ_y are standard deviations of x, y. σ_{xy} is the covariance between x, y. Each luminance, contrast and structure components are range between 0 and 1. Being closest to 1 means that windows x, y are like each other [28].

3.3.3. Bitrate

Bitrate refers to number of bits processed per unit of time. Unit of time is determined by the analyzer and seconds usually is preferred. Bitrate is affected by several components of the video; mainly the resolution and the frame per second (fps) are two biggest components that affect bitrate in video codecs.

Bitrate is the most important measurement tool in our daily life, most interactive media tools and real time videos use bitrate to adjust the quality of the video. YouTube is the most apparent example to most of us since most of the people watch YouTube. In YouTube the video quality is adjusted according to the maximum bitrate that the internet can be provided for the device. Therefore, changing qualities automatically (successfully maintaining maximum bitrate efficiency), YouTube provides the user uninterrupted visual experiment with maximum quality. Not just YouTube but most of the biggest video providers are trying to maximize the bitrate for each individual device, in order to increase the quality of experience while watching videos.

Codecs perform mathematical operations on especially in encoding part. Therefore, the heavier the calculations get, the more bitrate the codec needs. Generally low bitrates are associated with bad quality and high bit rates are associated with higher quality; however, bitrate itself solemnly does not give precise information about the video codec.

Both PSNR or SSIM and bitrate values alone are not giving full comparison between video codecs. In the experiment and result interpretation part, video codecs will be compared in terms of both quality metric and bitrate values. Most of the time under same bitrates better-quality values give better codec performances. Since it encodes same amount of video size in a specific time with better quality.

3.3.4. Bjontegaard Delta Rate (BD Rate)

BD Rate is a metric to calculate the relation between rate-distortion (RD) curves. BD Rate needs at least four points on RD curves to perform the calculation. It is mostly used in video processing for bitrate-quality comparisons of different RD curves. BD rate answers the following questions. BD rate outputs two information between two RD curves. First is the bitrate save between two RD curves in terms of percentage and second is the PSNR gain in dB. Both outputs help us to make deductions between RD curves. BD rate for PSNR acquired as follows, first for every point the fitted distortion polynomial $D(R)$ is calculated.

$$D(R) = a\,(\log R)^3 + b\,(\log R)^2 + c\,(\log R)^1 + d$$

Where R is the bitrate and a, b, c, d are the parameters of the fitting function calculated such that the fitted polynomial passes through all points. Next step is averaging difference between two RD curves PSNR. The equation is:

$$\Delta D = \frac{1}{r_{max} - r_{min}} \int_{r_{min}}^{r_{max}} [D_2(r) - D_1(r)]\, dr$$

Where $D_2(R)$ and $D_1(R)$ are fitted distortion polynomials, r_{max} is the minimum bitrate point of both RD curves maximum bitrate points and r_{min} is the maximum of both RD curves minimum bitrate points [31].

BD rates are used for all comparisons in this thesis scope. Figure 3.2 shows the example of how to interpret the output BD Rates for simple bitrate and PSNR values.

Bitrate	PSNR	Bitrate	PSNR		
100	40	50	40		
150	42	75	42	BDBR	-50
200	44	100	44	BDPSNR	4,93
250	46	125	46		
300	48	150	48		

Bitrate	PSNR	Bitrate	PSNR		
100	40	100	42		
150	42	150	44	BDBR	-23,8
200	44	200	46	BDPSNR	2,00
250	46	250	48		
300	48	300	50		

Figure 3.2. Example of BD Rate

3.3.5. Encoding Time

Encoding time is an important aspect when real time applications and video streaming platforms are being considered. On demand platforms do not depend entirely on encoding time process, they are more obligated to better bitrate savings and quality values. However, in real time application of video industry saving time and minimizing delay between end users are important, therefore the codecs are compared between and inside with different codec parameters. Drastic differences in encoding time results in bad codec settings for the end user.

3.4. Experimental Parameters in Video Codecs

Within the scope of this thesis, there are several different encoding parameters compared with using the three codecs. These parameters are the most crucial parameters of the encoding process. Moreover, they are set to different values and the resulting encoding process efficiency changes will be studied in the experimental part of the thesis.

3.4.1. Coding Unit Size

After the HEVC was introduced, the macroblock concept became history. The HEVC introduced coding unit system into video coding industry. It is not distinctive from macroblock concept; however coding units' size can be changeable. These coding units can be partitioned into smaller coding units, followed by some specific rules specified by codecs. Codecs can be specified by the maximum coding unit size that will be used in the encoding process. This maximum coding unit size will determine the number of the blocks that an image will be partitioned. Stationary parts in the image, meaning that there is not much of a difference in YUV spec of the coding unit, does not get partitioned into smaller parts.

3.4.2. Maximum Partition Depth

Maximum partition depth described in Part 2.2.1 is the decision of how deep a single coding unit is going to be partitioned in the encoder. It is one of the most important features to decide on, because the most time-consuming step is usually block partitioning part. Encoders have their own decision algorithms in order to decide on maximum partition depth [32]. These algorithms usually calculate the similarities between neighboring coding units. If there is enough difference between neighboring coding units after calculations, the coding unit is partitioned into different coding units [33]. More coding units eventuate in better coding efficiency and save bitrate. However, this also brings high encoding complexity and competition burden [34]. Therefore, deciding the maximum partition depth according to the intended purpose and available setup is crucial for the maximum encoding efficiency. Figure 3.3 shows maximum partition depth and coding units work. Grass areas are not partitioned at all since all pixels in coding unit are almost the same values and variance through the coding unit is close to zero. However, horseman parts of the image are partitioned into smaller and smaller parts to not lose any information when intra and inter predictions start.

Figure 3.3. Optimal CU representation of Sequence RaceHorses (WVGA) optimized by HEVC [35]

3.4.3. GOP size and I-Frame interval

Group of Pictures (GOP) consists of consecutive frames. GOP contains three different frame types, namely I-frame, B-frame and P-frame [36].

I-frame is a complete image, it is still compressed under certain steps of the encoder. However, it contains all the information that needs to be reconstructed at the decoder side, only fully by itself. Every GOP starts with an I-frame, this I-frame is being used by P and B frames. Moreover, I-frame contains more data than other frames.

B and P frames are called representative frames as well. P-frames are referenced from past frames and B-frames are referenced from past and future frames inside the GOP [37]. Figure 3.4 shows the relations between I, P and B frames.

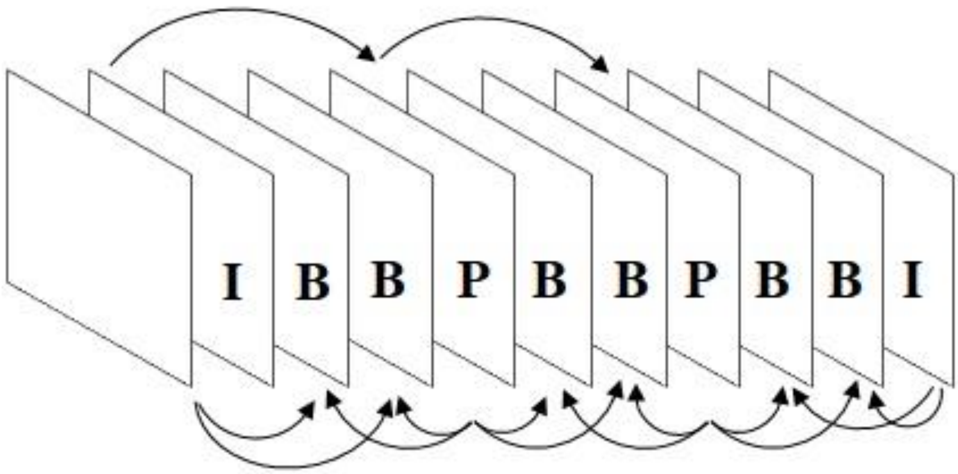

Figure 3.4. GOP frames [37]

3.4.4. Quantization Parameter (QP)

Quantization process is changing the continuous time signal into discrete numbers to represent the original analog signal in digitalized series of number. Throughout this process this analog data can be processed and used by the encoder. Quantization process is controlled by a QP. In this process the redundant data is discarded, in other words QP decides on how much spatial data is going to be saved while compressing the video [38]. Since these encoders are lossy encoders, meaning that the information is not always hundred percent saved into the encoder side, the information in the compressed video is sacrificed in quantization step. In figure 3.2 lower QP means that more detail is saved and transferred to the decoder side. However, bit rate increases by increasing quality level of the encoded video as well. Higher bitrate means that more information must be send through the decoder side. Higher QP results more relevant data to be discarded and the resulting PSNR will decrease at the decoder side

[39]. Therefore, finding a QP is challenging for the encoders, since the video characteristics are important to decide on using high QP or low. Within the scope of this thesis, QP is used for creating RD curves and making comparisons between different CTU difference or I frame difference characteristics.

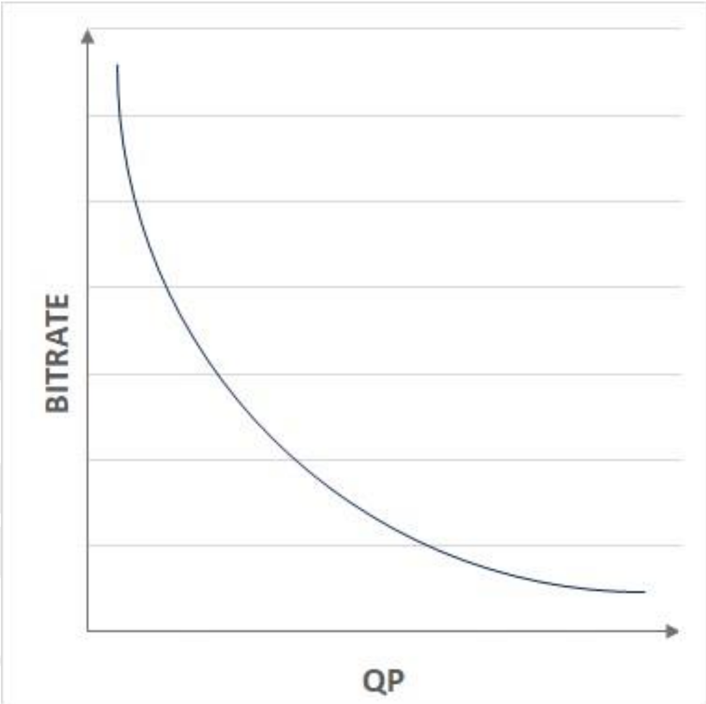

Figure 3.5. Bit rate – QP relation [39]

CHAPTER 4

EXPERIMENTAL RESULTS

4.1. Encoder Configurations

Encoders HEVC, AV1 and VVC are all configured to be able to perform random access encoding process. This process implements inserting intra predicted keyframes into the encoding process with regular intervals in order to enable play backing from specific encoding process time [21]. HEVC and VVC supports already prepared configuration files to the developers in order to use the encoder more conveniently. AV1 takes all the inputs by hand from command line and encoding is performed according to those input configurations.

4.1.1. HEVC Random Access Configuration

HEVC random access configuration file consists of the default configuration file. Parameters in the configuration file are listed in Appendix A. This configuration file is directly used in the encoding process. Besides configuration file, input file, frame rate, total frames, height and width of the videos are supplied from command line. Sample command line is:

"TAppEncoder.exe" -c encoder_randomaccess_main.cfg -f 600 -fr 60 -wdt 1024 -hgt 720 -i “input” -b “output”

4.1.2. VVC Random Access Configuration

VVC random access configuration file consists of the default configuration file. Parameters in the configuration file are listed in Appendix B. VVC supply same parameters as HEVC and sample command line is:

"TAppEncoder.exe" -c encoder_randomaccess_vtm.cfg -f 600 -fr 60 -wdt 1024 -hgt 720 -i “input” -b “output”

4.1.3. AV1 Encoding Options

AV1 does not support any pre-prepared configuration file, thus all the inputs should be provided by the user at the command line. These configurations are stated at Appendix C. For making the similar random-access process like HEVC and VVC some parameters are supplied as base from command line for all configurations [21].

Sample command line for constant quality configuration is:

"aomenc.exe" --min-partition-size=8 --max-partition-size=32 --cpu-used=0 --limit=150 --psnr --tune=psnr -b 8 --input-bit-depth=8 --threads=0 --profile=0 --width=3840 --height=2160 --fps=30/1 --lag-in-frames=19 --auto-alt-ref=1 --passes=2 --kf-max-dist=48 --kf-min-dist=48 --min-gf-interval=8 --max-gf-interval=8 --drop-frame=0 --static-thresh=0 --bias-pct=50 --minsection-pct=0 --maxsection-pct=2000 --arnr-maxframes=7 --arnr-strength=5 --sharpness=0 --undershoot-pct=100 --overshoot-pct=100 --frame-parallel=0 --tile-columns=0 --end-usage=q --cq-level=20 “input” -o “output”

Sample command line for constant bit per second configuration is:

"aomenc.exe" --min-partition-size=8 --max-partition-size=32 --cpu-used=0 --limit=150 --psnr --tune=psnr -b 8 --input-bit-depth=8 --threads=0 --profile=0 --width=3840 --height=2160 --fps=30/1 --lag-in-frames=19 --auto-alt-ref=1 --passes=2 --kf-max-dist=48 --kf-min-dist=48 --min-gf-interval=8 --max-gf-interval=8 --drop-frame=0 --static-thresh=0 --bias-pct=50 --minsection-pct=0 --maxsection-pct=2000 --arnr-maxframes=7 --arnr-strength=5 --sharpness=0 --undershoot-pct=100 --overshoot-pct=100 --frame-parallel=0 --tile-columns=0 --end-usage=vbr --target-bitrate=1500 “input” -o “output”

4.2. Video Classification

In the scope of this thesis, 5 different videos are used in order to create diversity for different movement and spatial behaviors. Figure 4.1 shows the classifications of the videos. “Akiyo” is the simplest video among the others, which have low movement and low complexity. “Container” differs from class “Akiyo” with the higher complexity overall, but the temporal data is about the same, therefore same movement behavior with “Akiyo”. “Foreman” has higher movement behavior than both “Akiyo” and “Container”; however, “Foreman” has less complexity than “Container” and more complexity than class “Akiyo”. “Mobile” is the most complex video over all with almost 200 spatial information. Finally, “Bus” has the highest mobility behavior among all videos with the highest temporal information.

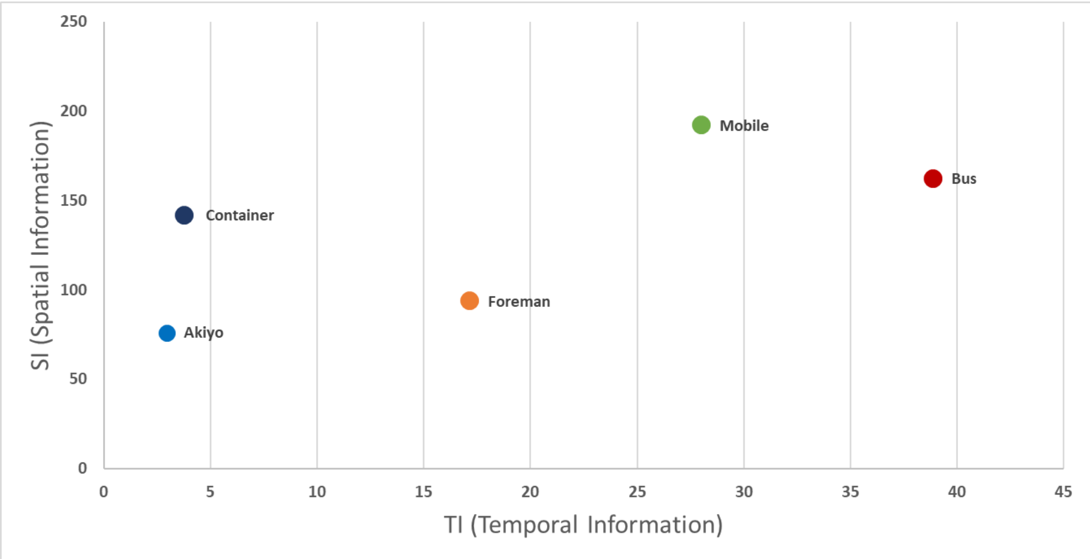

Figure 4.1. Video specifications with SI and TI information

4.3. CTU Size and Depth Difference

4.3.1. HEVC CTU and Depth Sizes

For HEVC CTU and depth sizes are adjusted. In the configuration file “MaxCUWidth” and “MaxCUHeight” parameters are adjusted 64 and 32, moreover “MaxPartitionDepth” parameter is set both 4 and 3 for 64x64 configuration and it is set to 3 for 32x32 configuration. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8. Quantization parameter is set to 20, 26, 32, 38, 44 while performing the tests. I frame interval is fixed to 32. Rate control is off. Other parameters are all taken as default parameters.

4.3.1.1. HEVC Akiyo

Table 4.1. *HEVC Akiyo PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	45.18	42.39	39.18	35.76	32.02
CTU Size 64 Depth 3	45.12	42.30	39.09	35.64	31.92
CTU Size 32 Depth 3	45.20	42.36	39.15	35.66	31.96

Table 4.2. *HEVC Akiyo SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	0.99	0.98	0.97	0.95	0.90
CTU Size 64 Depth 3	0.99	0.98	0.97	0.94	0.90
CTU Size 32 Depth 3	0.99	0.98	0.97	0.94	0.90

Table 4.3. *HEVC Akiyo bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	189.77	89.01	45.68	25.54	15.12
CTU Size 64 Depth 3	195.07	91.22	46.64	25.99	15.17
CTU Size 32 Depth 3	190.49	89.43	45.85	25.74	15.24

Figure 4.2 illustrates the RD curves of video Akiyo for different CTU size and depth configurations, RD curves are very similar to each other because the effect of changing CTU sizes and depths for video Akiyo is very marginal. However when Table 4.1,

Table 4.2 and Table 4.3 are analyzed in terms of both saving the maximum amount of saving bitrate and best PSNR results are acquired by CTU size 64 depth 4 configuration. This is very expected, because the widest possibility of CTU size and minimum CU size is acquired by this configuration; therefore, it should provide the best performance. CTU size 32 depth 3 configuration is better than the CTU size 64 depth 3 configuration, this is due to for lower resolution videos 64x64 CTU size is too big and it merely used by the codec configuration; therefore accessing smallest size CU (4x4) is more important for efficient coding. On the other hand, CTU size 64 depth 3 configuration can only partition the CTU sizes up to 8x8 CU.

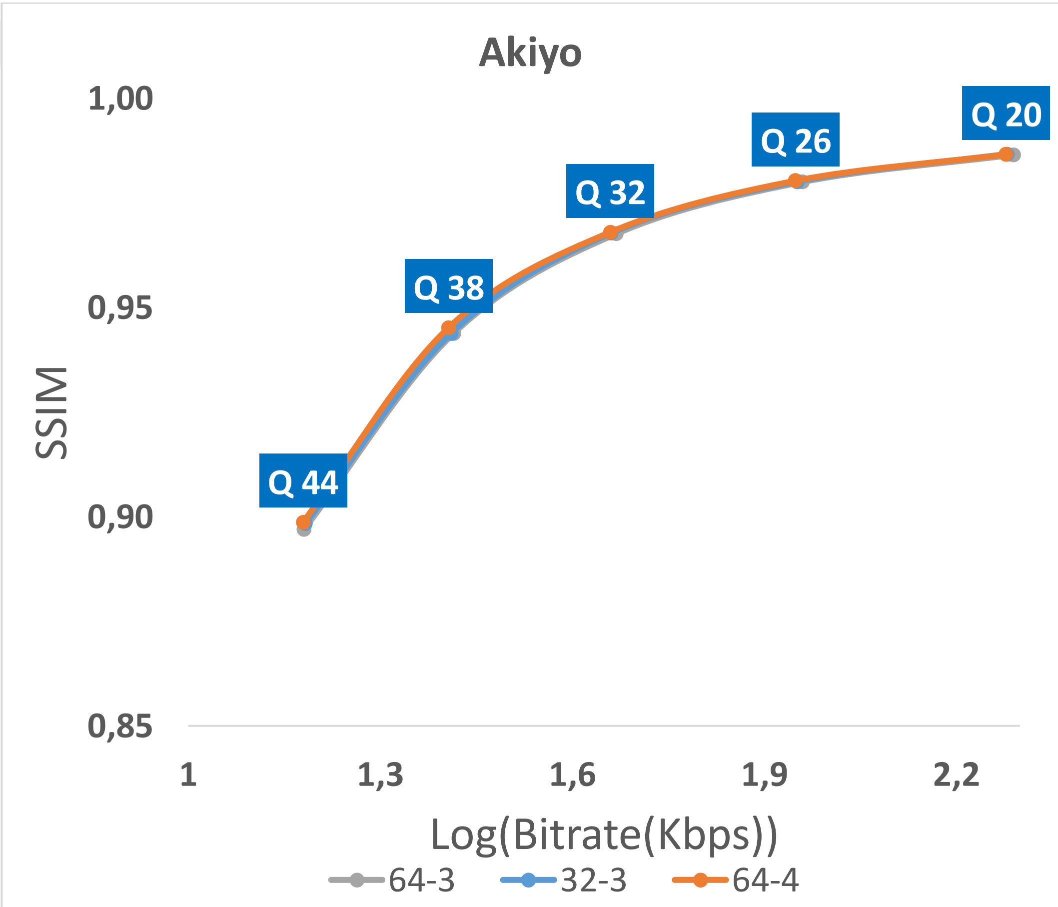

Figure 4.2. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.1.2. HEVC Bus

Table 4.4. *HEVC Bus PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	39.67	35.41	31.33	27.73	24.68
CTU Size 64 Depth 3	39.58	35.26	31.14	27.56	24.59
CTU Size 32 Depth 3	39.69	35.40	31.32	27.71	24.63

Table 4.5. *HEVC Bus SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	0.98	0.97	0.92	0.83	0.71
CTU Size 64 Depth 3	0.98	0.96	0.92	0.83	0.71
CTU Size 32 Depth 3	0.98	0.97	0.92	0.83	0.71

Table 4.6. *HEVC Bus bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	1603.87	713.46	309.25	137.83	62.66
CTU Size 64 Depth 3	1670.09	749.00	323.13	140.01	62.90
CTU Size 32 Depth 3	1603.74	711.94	310.06	138.49	63.44

Figure 4.3 shows the RD curves of video BUS for different CTU size and depth configurations, again like video Akiyo the results are close to each other. One main difference is that from the Figure 4.3 it can be actually seen that CTU size 64 with

depth 3 configuration performance is worse than other configurations. Moreover, when Table 4.4, Table 4.5 and Table 4.6 are analyzed the gap between CTU size 64 and depth 3 configuration and other 2 configurations increase. Because the Bus video has the maximum temporal difference, and by increasing the temporal behavior of the video, the codec needs to use the minimum size CU more for better encoding efficiency. Therefore, CTU size 64 depth 3 configuration performs worse than other 2 configurations.

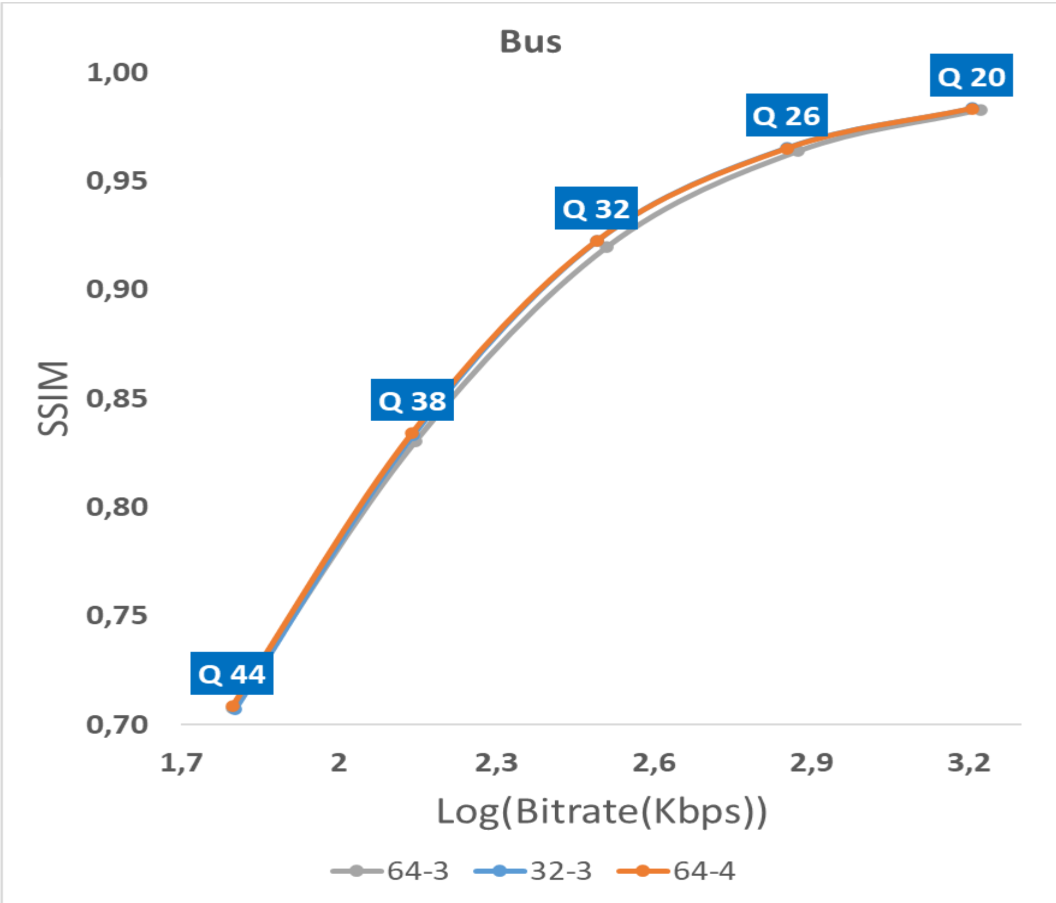

Figure 4.3. Bus SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.1.3. HEVC Container

Table 4.7. *HEVC Container PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	42.17	38.87	35.60	32.35	28.94
CTU Size 64 Depth 3	42.11	38.80	35.51	32.26	28.85
CTU Size 32 Depth 3	42.17	38.88	35.58	32.32	28.89

Table 4.8. *HEVC Container SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	0.97	0.95	0.91	0.87	0.83
CTU Size 64 Depth 3	0.97	0.95	0.91	0.87	0.83
CTU Size 32 Depth 3	0.97	0.95	0.91	0.87	0.82

Table 4.9. *HEVC Container bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	414.89	181.86	84.71	43.08	22.63
CTU Size 64 Depth 3	421.39	185.26	86.11	43.39	22.79
CTU Size 32 Depth 3	416.47	182.56	84.73	43.12	22.73

Similar to video Akiyo, video Container has very close RD curves as it can be seen from Figure 4.4. When the values in Table 4.7, Table 4.8 and Table 4.9 are compared again CTU size 64 depth 3 configuration performs worse than other 2 configurations;

while other 2 configurations performs close to each other. One important thing to mention is that the efficiency increases for CTU size 64 and depth 4 configuration when the QP increases. CTU size 64 and depth 4 configuration and CTU size 32 depth 3 configurations results almost same PSNR outputs when QP is 20. However, when QP is increased to 44, the codec needs to find more efficient ways of transferring the data; therefore, it uses bigger CTU sizes more often. When QP becomes smaller, the codec has more bitrate to work with, therefore does not need bigger CTU sizes that frequently. Therefore, 0.05 PSNR difference occur between these 2 configurations tested at 44 QP parameter level. This behavior can be seen for all videos.

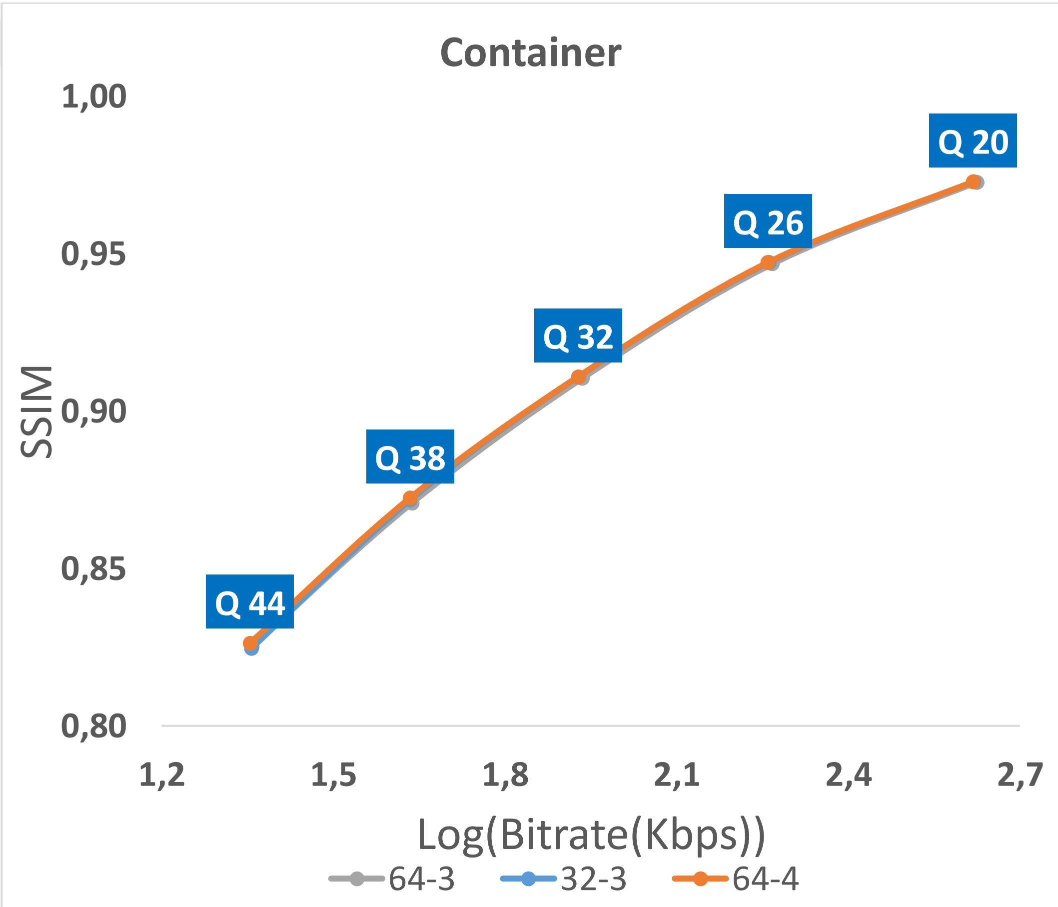

Figure 4.4. Container SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.1.4. HEVC Foreman

Table 4.10. *HEVC Foreman PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	41.64	38.42	35.00	31.67	28.72
CTU Size 64 Depth 3	41.50	38.20	34.81	31.57	28.59
CTU Size 32 Depth 3	41.65	38.39	34.94	31.60	28.46

Table 4.11. *HEVC Foreman SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	0.98	0.96	0.92	0.87	0.81
CTU Size 64 Depth 3	0.98	0.96	0.92	0.87	0.81
CTU Size 32 Depth 3	0.98	0.96	0.92	0.87	0.81

Table 4.12. *HEVC Foreman bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	631.69	264.98	116.36	53.96	27.80
CTU Size 64 Depth 3	668.18	275.60	119.34	54.78	27.23
CTU Size 32 Depth 3	633.47	266.05	116.64	55.17	28.05

Figure 4.5 demonstrates RD curves for video Foreman under different CTU size and depth configurations. Results are similar with other videos, CTU size 64 and depth 3 configuration performs worse in terms of both bitrate and quality; while Table 4.10,

Table 4.11, and Table 4.12 shows the numbers to support this. CTU size 64 depth 4 configuration and CTU size 32 depth 3 configurations perform very similar at low QP values; however again when QP value gets higher the performance of CTU size 64 and depth 4 becomes better than CTU size 32 depth 3 configuration.

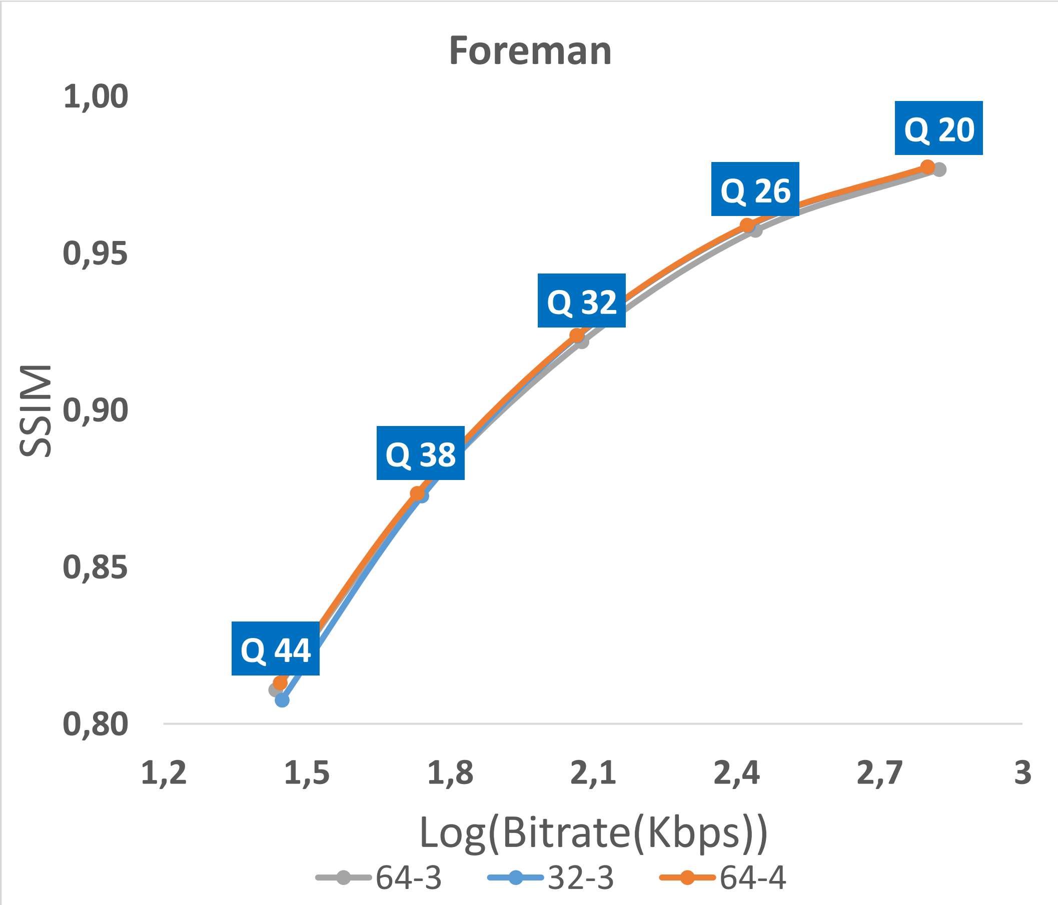

Figure 4.5. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.1.5. HEVC Mobile

Table 4.13. *HEVC Mobile PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	39.24	34.33	30.48	27.36	23.98
CTU Size 64 Depth 3	39.15	34.22	30.33	27.16	23.77
CTU Size 32 Depth 3	39.25	34.34	30.49	27.37	23.94

Table 4.14. *HEVC Mobile SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	0.99	0.98	0.96	0.91	0.81
CTU Size 64 Depth 3	0.99	0.98	0.95	0.91	0.81
CTU Size 32 Depth 3	0.99	0.98	0.96	0.91	0.81

Table 4.15. *HEVC Mobile bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	2142.04	842.85	339.11	166.71	84.88
CTU Size 64 Depth 3	2206.23	879.71	356.90	173.35	86.94
CTU Size 32 Depth 3	2143.08	843.85	339.89	167.64	85.97

Like other low-resolution videos Mobile also performs worse when CTU size 64 and depth 3 configuration is used, it can be seen from Figure 4.6. Mobile video has the most SI amongst CIF quality videos. Since it has the most complexity, the bitrate values are the highest as expected. From Table 4.15 it can be seen that for QP value 20 the bitrate to encode the video passes 2 MBps. Again, CTU size 64 depth 4 configurations performs best in terms of both quality output and bitrate.

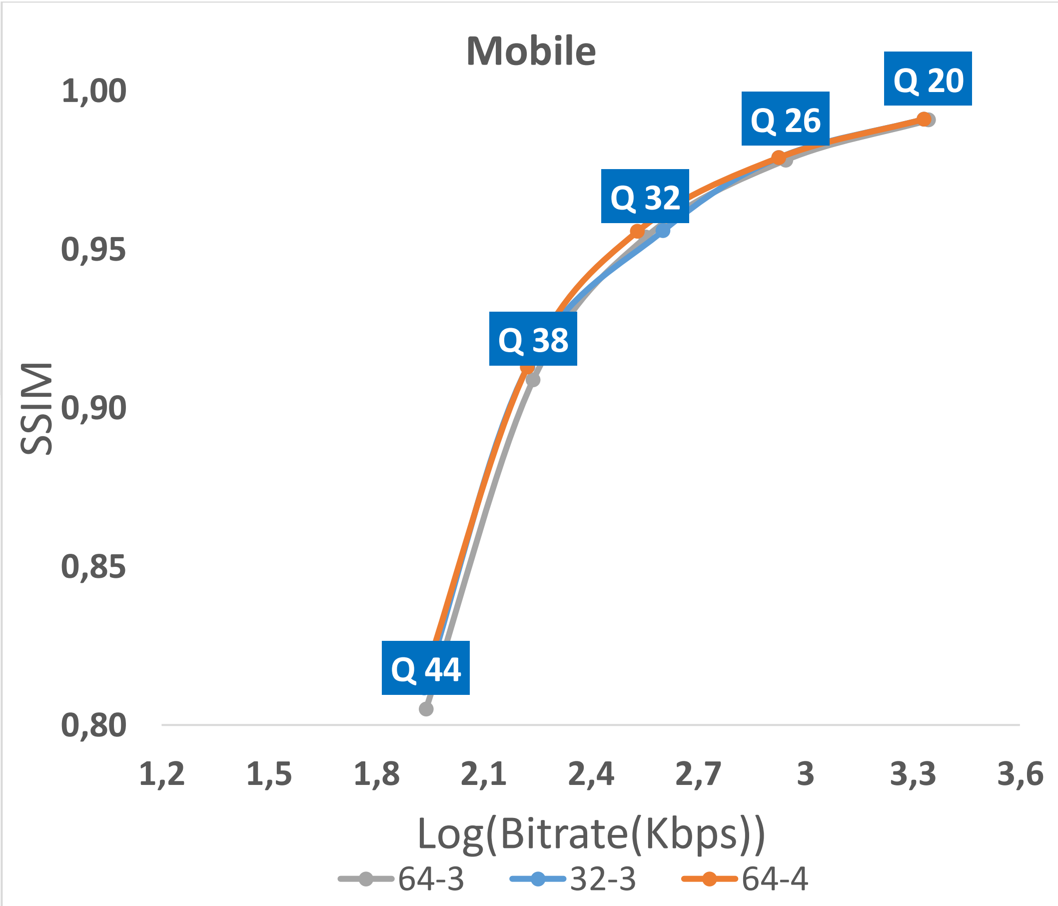

Figure 4.6. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.1.6. HEVC Basketball

Table 4.16. *HEVC Basketball PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	47.67	45.99	43.36	40.17	37.03
CTU Size 64 Depth 3	47.67	45.99	43.35	40.17	37.03
CTU Size 32 Depth 3	47.55	45.76	43.03	39.86	36.63

Table 4.17. *HEVC Basketball SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	0.99	0.99	0.98	0.98	0.96
CTU Size 64 Depth 3	0.99	0.99	0.98	0.98	0.96
CTU Size 32 Depth 3	0.99	0.99	0.98	0.97	0.96

Table 4.18. *HEVC Basketball bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 64 Depth 4	10527.68	3577.82	1716.91	905.76	513.79
CTU Size 64 Depth 3	10145.35	3609.54	1723.76	906.47	514.07
CTU Size 32 Depth 3	8791.72	3199.91	1583.98	872.02	532.84

There is a significant performance difference between Basketball video and other CIF quality videos. Figure 4.7 shows RD curves for video Basketball and for QP value 44, it is seen that how bad CTU size 32 depth 3 configuration performs than other 2 configurations. When the QP increases, the gap between all configurations are started to diminish. This shows the importance of having a bigger size CTU especially when higher resolution videos are encoded by HEVC. Table 4.16, Table 4.17 and Table 4.18 provides the numbers and for QP 44 case, there is 0.4 PSNR difference between CTU size 32 depth 3 configuration and other 2 configurations, while also CTU size 32 depth 3 configuration needs almost 20 kBps more than other 2 configurations. Other important thing for video Basketball is that for all QP levels except 44, CTU size 32 with depth 3 configuration performs lower bitrate as well as lower quality levels. This indicates how frequently the 64x64 CTU size is used for video Basketball since it has higher dimensions than CIF videos (3840x2160), and 64x64 CTU size becomes critical for codec to efficiently code the video.

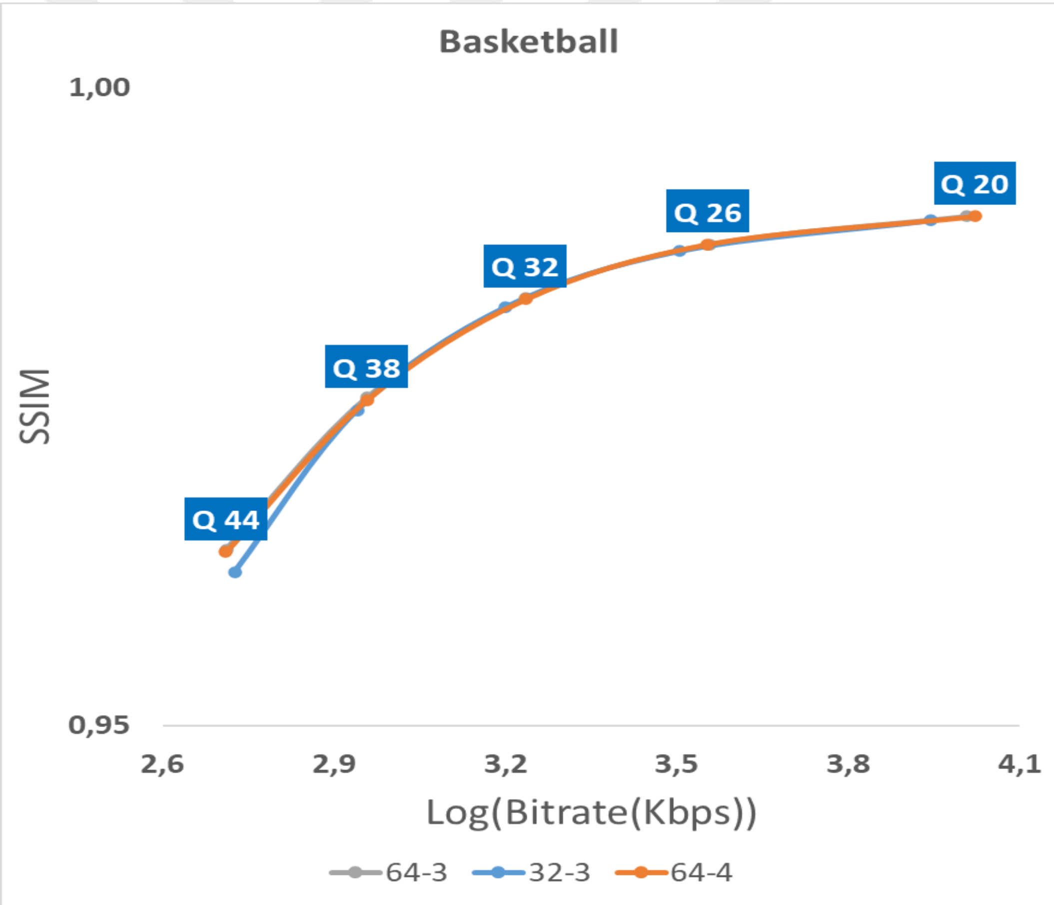

Figure 4.7. Basketball SSIM vs Log(Bitrate(Kbps) graph for different CTU and depth sizes

4.3.1.7. HEVC BD Rate results

HEVC CTU results are very close to each other in most of the cases; however, there are two main aspects to consider for the results. As expected, the best results are acquired when the CTU size is bigger as well as the depth is maximum, which is 64x64 CTU size configuration with maximum partition depth of 4. This configuration has the widest possibilities among other two configurations. As BD rate results at Figure 4.8 show that 64x64 with depth 4 configuration always superior than other two. One important thing to look at here is that when the complexity of the videos is higher, 64x4 depth 4 configuration become even more superior than other two.

When 64x64 depth 3 configuration is compared against the other two configurations, the lowest performance acquired for CIF quality videos. As in the case of the comparison between 64x64 depth 4 and 32x32 depth 3 configurations, when the complexity of the videos increases, the difference between 64x64 depth 3 and other two configurations become more surplus. In video Container 64x64 depth 3 configuration needs %3,18 more bitrate when compared to 64x64 depth 4 configuration and %2,56 more bitrate when compared to 32x32 depth 3 configuration at the same quality level. At the same bitrate level 64x64 depth 3 configuration results 0,14 PSNR less when compared to 64x64 depth 4 configuration and 0,12 less when compared to 32x32 depth 3 configuration. For a more complex video Bus the results become steeper. 64x64 depth 3 configuration needs %7,15 more bitrate when compared to 64x64 depth 4 configuration and %6,55 more bitrate when compared to 32x32 depth 3 configuration at the same quality level. At the same bitrate level 64x64 depth 3 configuration results 0,32 PSNR less when compared to 64x64 depth 4 configuration and 0,29 less when compared to 32x32 depth 3 configuration.

When 4K video Basketball is compared with the remaining CIF videos, the results are not entirely same when compared to CIF results. For CIF videos 32x32 depth 3 configurations is always better than the 64x64 depth 3. Therefore, for CIF videos going to the smallest CU is important, while bigger CU sizes are not usually used, because the video sizes are small as well, therefore 64x64 CU size is rarely used for encoding process. On the other hand, in 4K video 64x64 depth 3 configuration is better than 32x32 depth 3. This is because the complexity of the frame. For 4K video going to the smallest possible CU size is not feasible, using bigger CU size for smoother regions is more important. Therefore, as the technology increases and resolution of the videos become larger and larger for more efficient video encoding, CTU sizes should also increase. The results are supporting this idea as well. Even though the number difference is too small for Basketball, 64x64 depth 4 configuration is better than 64x64 depth 3 configuration by %0.10 bitrate save at the same quality level and 0,02 dB PSNR improvement at the same bitrate level. Moreover, 64x64 depth 3 configuration is better than 32x32 depth 3 configuration by %0.15 bitrate save at the same quality level and 0,02 dB PSNR improvement at the same bitrate level.

Overall, for HEVC when lower resolution videos are taken into consideration, selecting the smaller CTU size with same partitioning depth is better, while for higher resolution videos selecting maximum CTU size is more important than choosing maximum partitioning depth.

Video	CTU-Depth	Metric	64-4	64-3	32-3
Akiyo	64-4	BDBR		3,86	1,42
		BDPSNR		-0,19	-0,07
	64-3	BDBR	-3,71		-2,33
		BDPSNR	0,19		0,12
	32-3	BDBR	-1,40	2,39	
		BDPSNR	0,07	-0,12	
Bus	64-4	BDBR		7,15	0,53
		BDPSNR		-0,32	-0,02
	64-3	BDBR	-6,67		-6,15
		BDPSNR	0,32		0,29
	32-3	BDBR	-0,53	6,55	
		BDPSNR	0,02	-0,29	
Container	64-4	BDBR		3,18	0,59
		BDPSNR		-0,14	-0,03
	64-3	BDBR	-3,08		-2,50
		BDPSNR	0,14		0,12
	32-3	BDBR	-0,59	2,56	
		BDPSNR	0,03	-0,12	
Foreman	64-4	BDBR		6,74	2,26
		BDPSNR		-0,27	-0,09
	64-3	BDBR	-6,31		-4,08
		BDPSNR	0,27		0,17
	32-3	BDBR	-2,21	4,25	
		BDPSNR	0,09	-0,17	
Mobile	64-4	BDBR		7,42	4,75
		BDPSNR		-0,34	-0,22
	64-3	BDBR	-6,91		-2,46
		BDPSNR	0,34		0,12
	32-3	BDBR	-4,53	2,52	
		BDPSNR	0,22	-0,12	
Basketbal	64-4	BDBR		0,10	0,30
		BDPSNR		-0,02	-0,04
	64-3	BDBR	-0,10		0,15
		BDPSNR	0,02		-0,02
	32-3	BDBR	-0,30	-0,15	
		BDPSNR	0,04	0,02	

Figure 4.8. HEVC BD Rates of all videos for different CTU and depth sizes

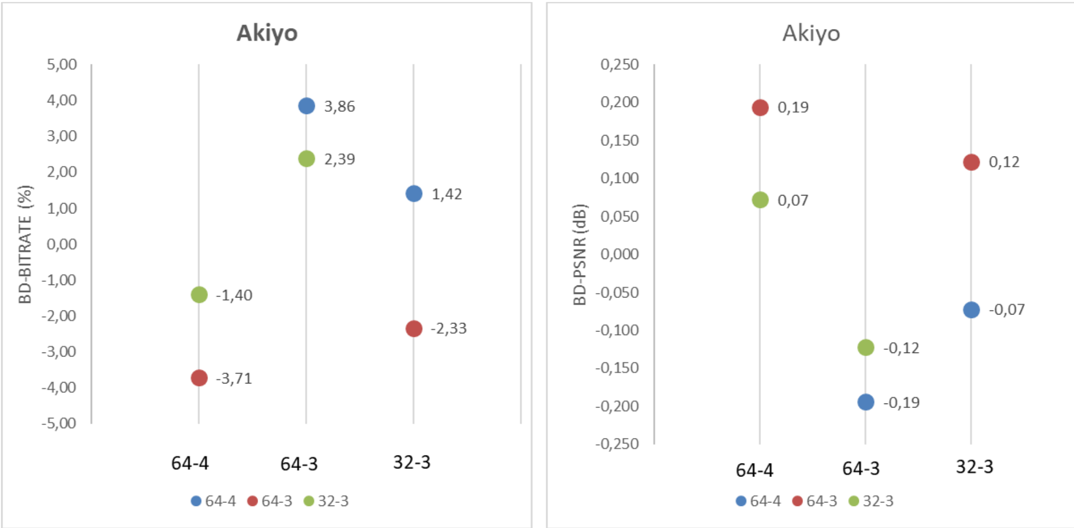

Figure 4.9. Akiyo BD Rates for HEVC

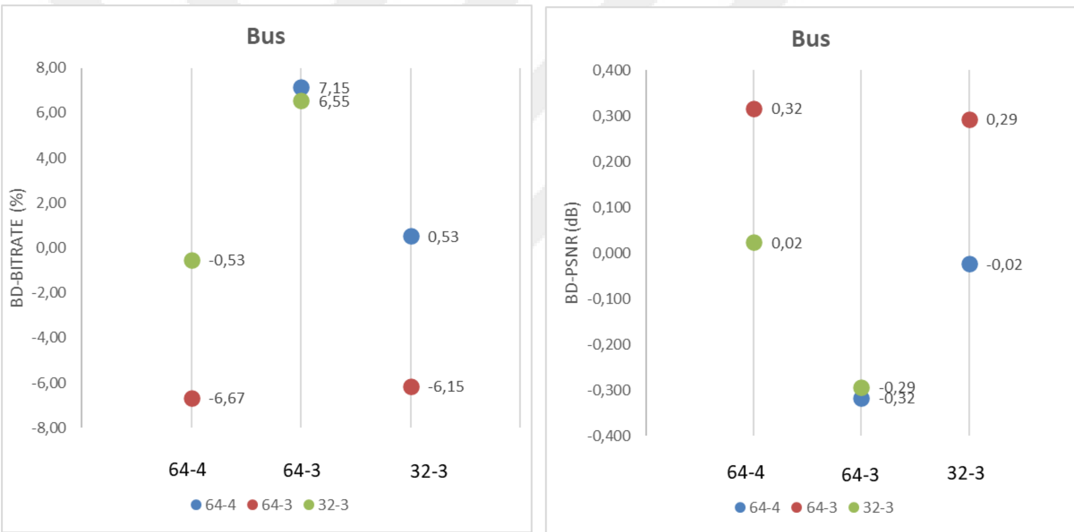

Figure 4.10. Bus BD Rates for HEVC

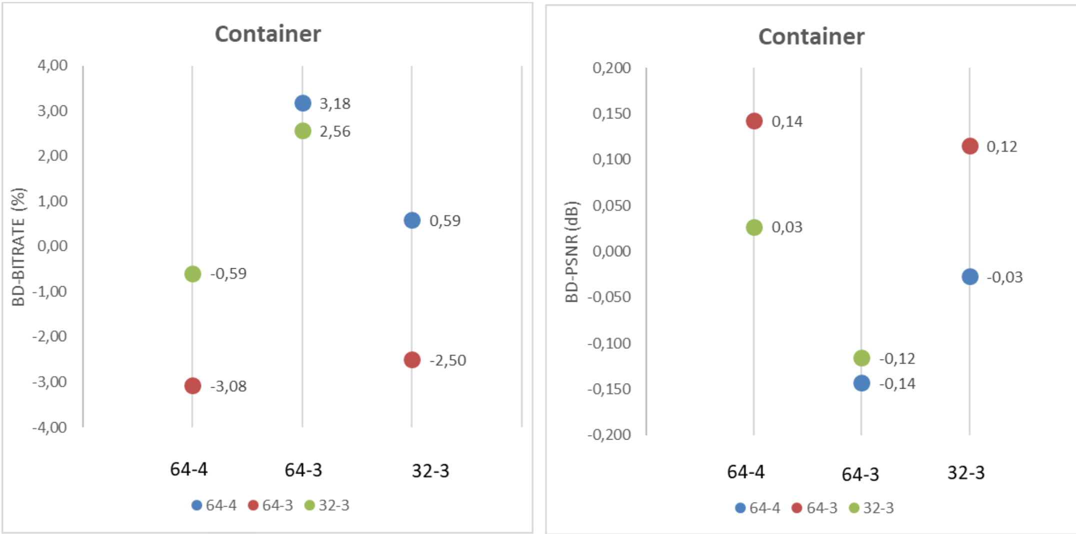

Figure 4.11. Container BD Rates for HEVC

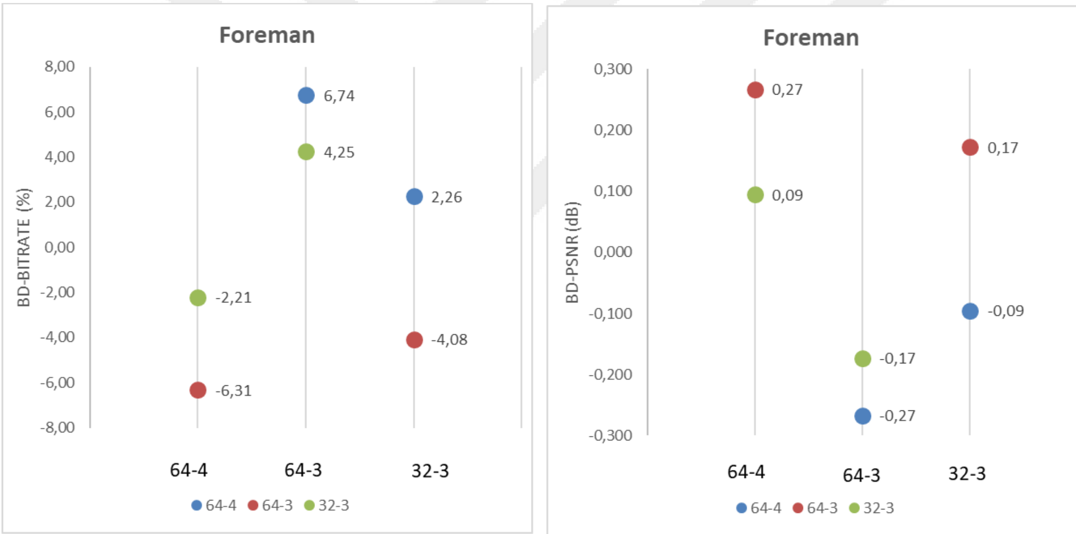

Figure 4.12. Foreman BD Rates for HEVC

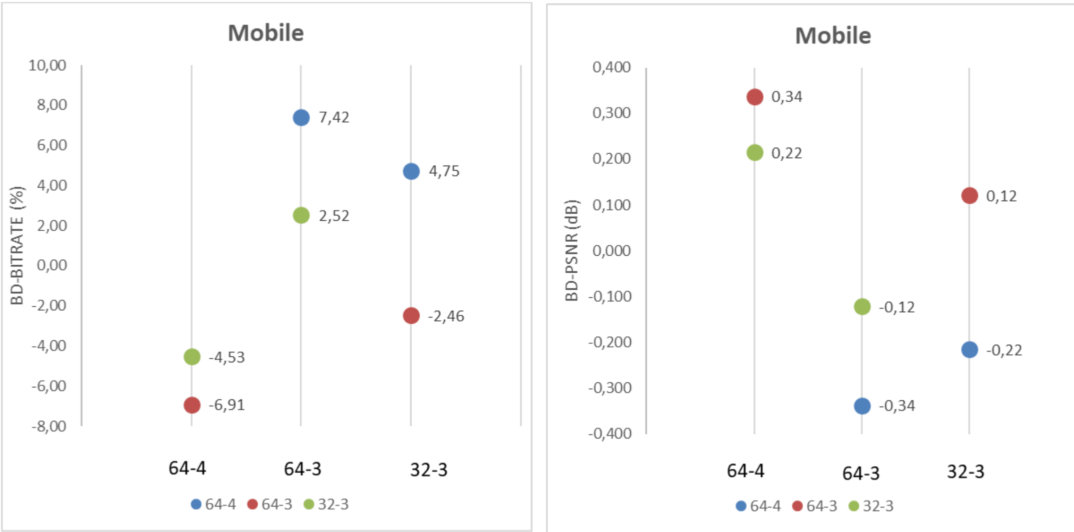

Figure 4.13. Mobile BD Rates for HEVC

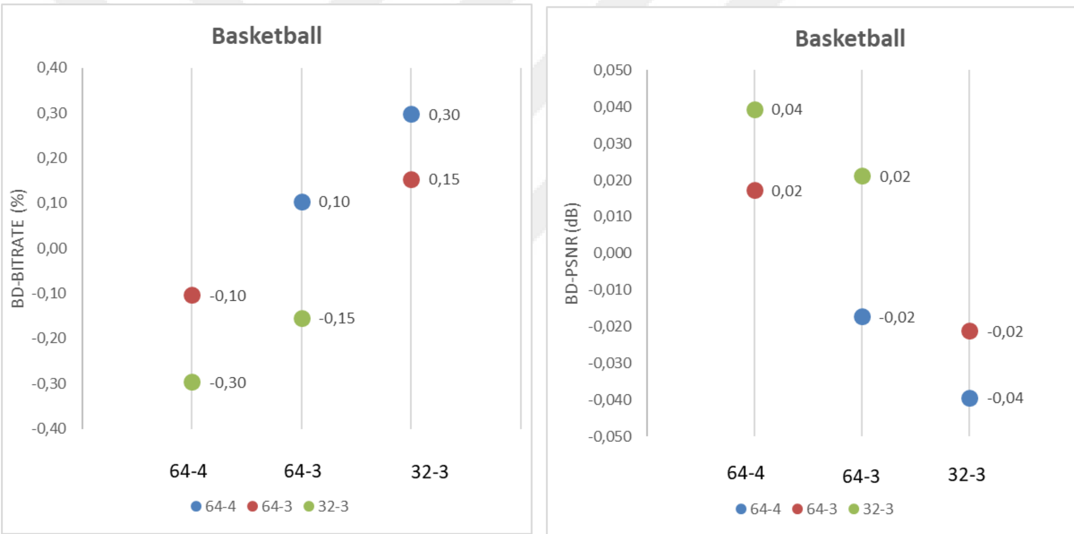

Figure 4.14. Basketball BD Rates for HEVC

4.3.2. VVC CTU and Depth Sizes

In the configuration file “CTUSize” parameter is set to 128, 64 and 32. While setting “CTUSize” is set to 128 “HierarchyDepth” parameters are set at 3, for “CTUSize” 64 “HierarchyDepth” parameters are set at 3 and 4, finally for “CTUSize” 64 “HierarchyDepth” parameters are set at 3. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8. Quantization parameter is set to 20, 26, 32, 38, 44 while performing the tests. “IntraPeriod” is set to 32. Rate control is off. Other parameters are all taken as default parameters.

4.3.2.1. VVC Akiyo

Table 4.19. *VVC Akiyo PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	44.95	42.77	39.93	36.82	33.43
CTU Size 64 Depth 4	44.98	42.79	39.98	36.83	33.45
CTU Size 64 Depth 3	44.98	42.77	39.96	36.79	33.47
CTU Size 32 Depth 3	45.00	42.77	39.87	36.68	33.30

Table 4.20. *VVC Akiyo SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	0.99	0.98	0.97	0.96	0.92
CTU Size 64 Depth 4	0.99	0.98	0.97	0.96	0.92
CTU Size 64 Depth 3	0.99	0.98	0.97	0.96	0.92
CTU Size 32 Depth 3	0.99	0.98	0.97	0.95	0.92

Table 4.21. *VVC Akiyo bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	167.02	83.91	42.71	24.37	15.64
CTU Size 64 Depth 4	167.89	83.67	43.16	24.56	15.74
CTU Size 64 Depth 3	168.79	84.02	43.10	24.59	15.77
CTU Size 32 Depth 3	172.27	85.51	44.15	25.67	16.56

In VVC 128x128 CTU size is used as an enhancement over HEVC. The results are very close to each other when QP is low. However, as it can be seen from Figure 4.15 when QP becomes higher CTU 32 depth 3 configuration's performance starts to decrease. Especially for QP 44 its bitrate is high as well as quality less than other configurations. This clarify that VVC places emphasis more on bigger sizes CTU than HEVC. Table 4.19, Table 4.20 and Table 4.21 gives the numerical results for video Akiyo. Since Akiyo has the lowest temporal behavior using low CU sizes is not necessary and this impacts the efficiency especially for higher QP values. CTU size 128 and CTU size 64 configurations are close to each other for lower quality videos.

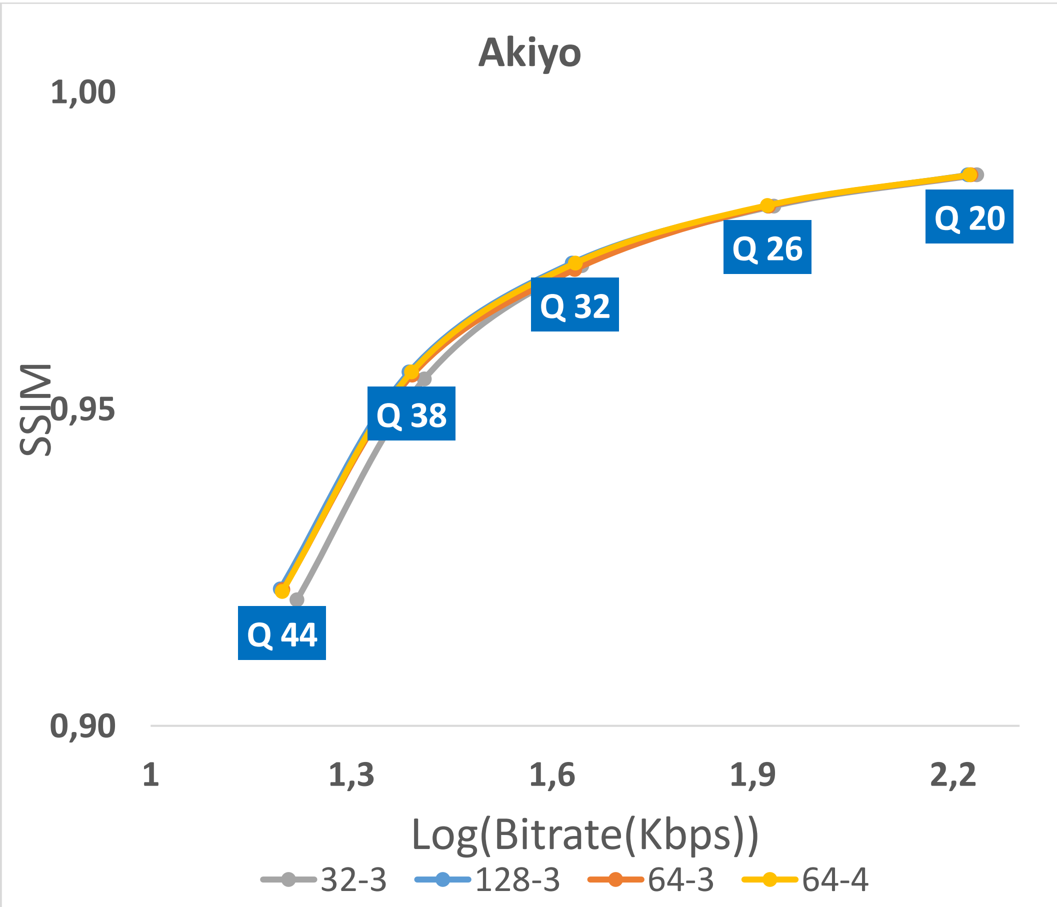

Figure 4.15. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.2.2. VVC Bus

Table 4.22. *VVC Bus PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	40.16	35.97	31.96	28.37	25.27
CTU Size 64 Depth 4	40.18	36.01	31.97	28.40	25.30
CTU Size 64 Depth 3	40.18	35.98	31.97	28.39	25.27
CTU Size 32 Depth 3	40.20	36.01	31.96	28.34	25.17

Table 4.23. *VVC Bus SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	0.98	0.97	0.93	0.85	0.73
CTU Size 64 Depth 4	0.98	0.97	0.93	0.86	0.74
CTU Size 64 Depth 3	0.98	0.97	0.93	0.85	0.74
CTU Size 32 Depth 3	0.98	0.97	0.93	0.85	0.73

Table 4.24. *VVC Bus bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	1393.04	609.24	262.59	113.70	49.45
CTU Size 64 Depth 4	1384.16	606.44	260.56	113.23	49.53
CTU Size 64 Depth 3	1396.46	611.60	263.23	114.29	49.87
CTU Size 32 Depth 3	1400.48	615.98	265.36	116.58	51.74

Video Bus shows similar results to Akiyo. Figure 4.16 demonstrates RD curves for different configurations. Again, for lower QP values the significance of having higher CTU size is important. Table 4.22, Table 4.23 and Table 4.24 gives the numerical values and it is seen that for QP 20 CTU size 32 depth 3 configuration has the best PSNR output, while for QP 44 it has the worst PSNR output. This solidifies that VVC uses higher CTU sizes for encoding when the bitrate limitation becomes lower. By that VVC tries to increase encoding efficiency under limited bitrate values.

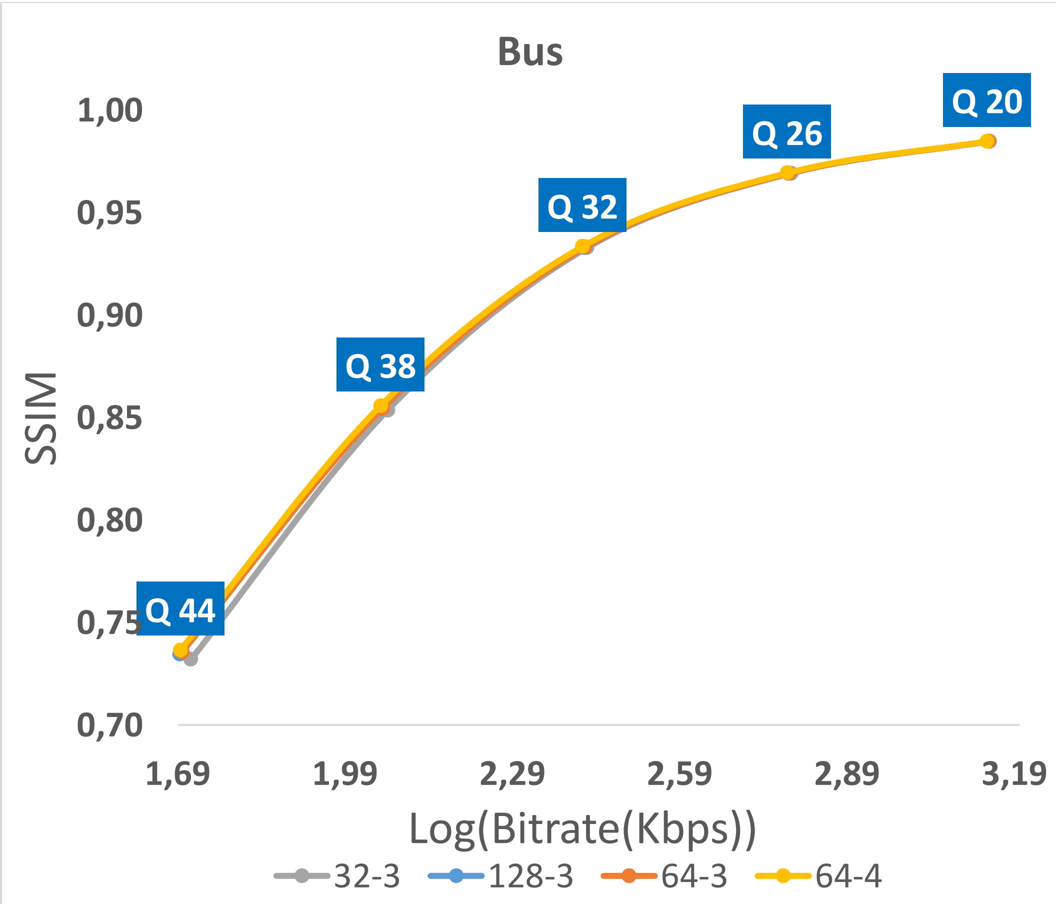

Figure 4.16. Bus SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.2.3. VVC Container

Table 4.25. *VVC Container PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	42.53	39.30	36.09	32.96	29.64
CTU Size 64 Depth 4	42.56	39.31	36.11	32.99	29.64
CTU Size 64 Depth 3	42.55	39.31	36.11	32.96	29.62
CTU Size 32 Depth 3	42.57	39.31	36.10	32.94	29.61

Table 4.26. *VVC Container SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	0.98	0.95	0.92	0.88	0.84
CTU Size 64 Depth 4	0.98	0.95	0.92	0.88	0.84
CTU Size 64 Depth 3	0.98	0.95	0.92	0.88	0.84
CTU Size 32 Depth 3	0.98	0.95	0.92	0.88	0.84

Table 4.27. *VVC Container bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	391.55	169.71	78.59	39.76	21.16
CTU Size 64 Depth 4	392.42	169.81	78.75	39.92	21.33
CTU Size 64 Depth 3	393.09	170.63	79.17	40.00	21.26
CTU Size 32 Depth 3	396.76	170.99	80.21	41.26	22.38

Container video has similar results with both Akiyo and Bus, as it can be seen from Figure 4.17. The results again close to each other, while for lower QP values having higher CTU size is more beneficial.

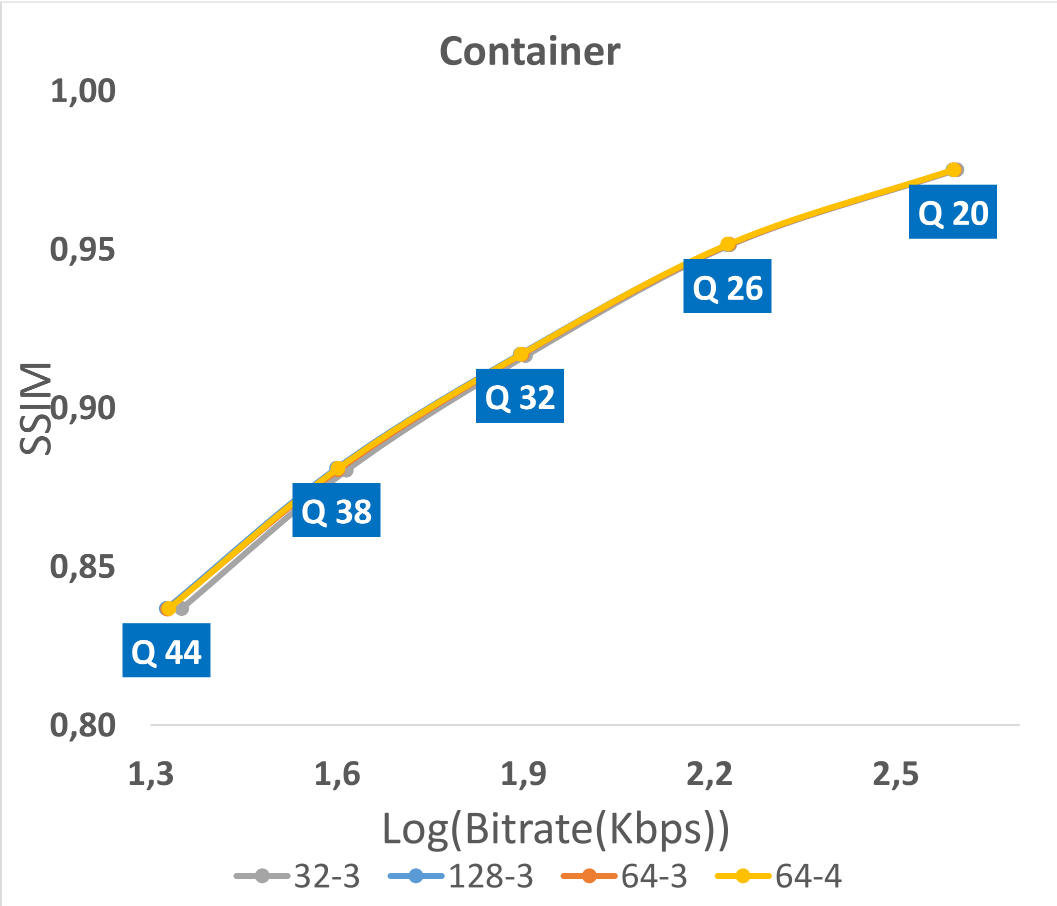

Figure 4.17. Container SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.2.4. VVC Foreman

Table 4.28. *VVC Foreman PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	41.69	38.76	35.52	32.37	29.43
CTU Size 64 Depth 4	41.75	38.77	35.50	32.41	29.40
CTU Size 64 Depth 3	41.74	38.76	35.50	32.37	29.40
CTU Size 32 Depth 3	41.74	38.68	35.39	32.19	29.15

Table 4.29. *VVC Foreman SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	0.98	0.96	0.93	0.89	0.83
CTU Size 64 Depth 4	0.98	0.96	0.93	0.89	0.83
CTU Size 64 Depth 3	0.98	0.96	0.93	0.89	0.83
CTU Size 32 Depth 3	0.98	0.96	0.93	0.88	0.83

Table 4.30. *VVC Foreman bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	521.24	223.26	97.33	45.57	23.60
CTU Size 64 Depth 4	520.72	222.04	96.93	45.82	23.61
CTU Size 64 Depth 3	525.19	223.96	97.36	45.65	23.54
CTU Size 32 Depth 3	533.66	228.10	100.18	47.97	25.31

Foreman video has also similar results with other videos. For Foreman video the performance decrease of CTU 32 depth 3 configuration can be seen more clearly from Figure 4.18. Table 4.28, Table 4.29 and Table 4.30 gives the numerical results. As in all cases of CIF videos CTU size 128x128 depth 5 configuration has the worst results for QP value 20 and best results for QP 44 in terms of PSNR quality.

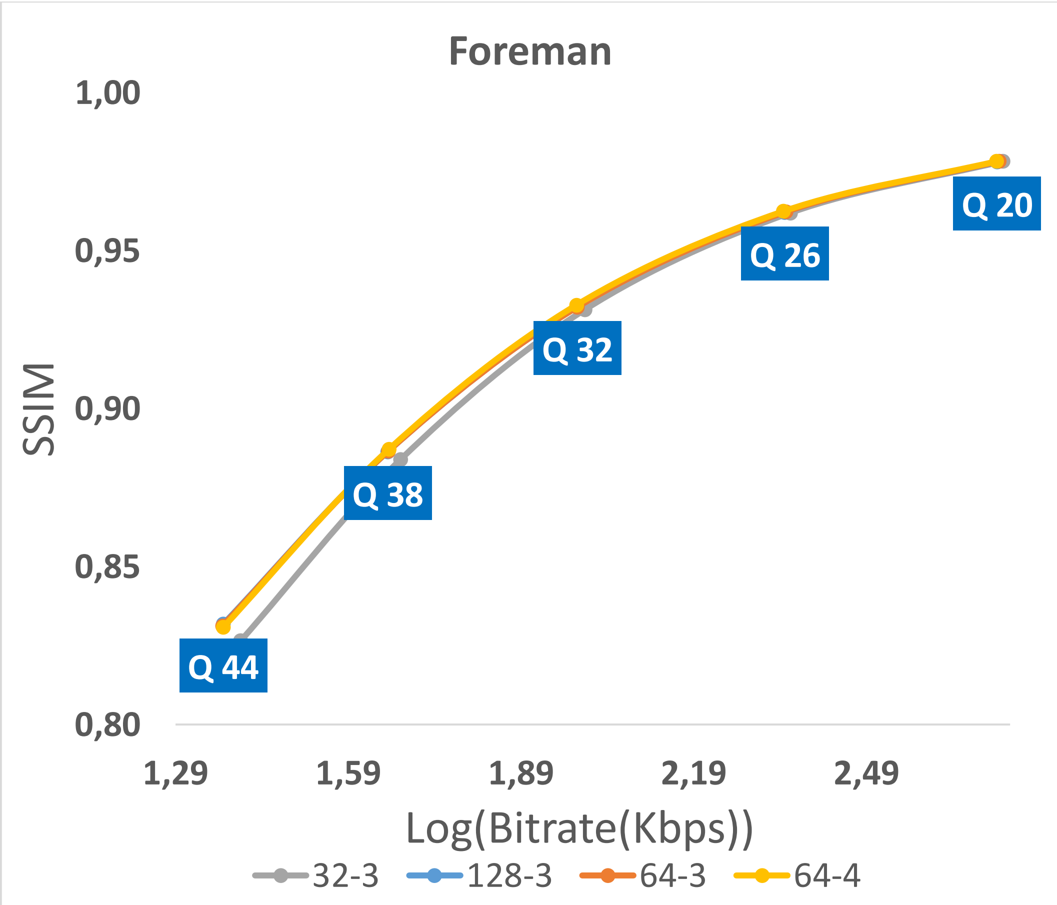

Figure 4.18. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.2.5. VVC Mobile

Table 4.31. *VVC Mobile PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	39.45	34.57	30.82	27.92	24.65
CTU Size 64 Depth 4	39.48	34.60	30.85	27.97	24.71
CTU Size 64 Depth 3	39.47	34.61	30.85	27.94	24.68
CTU Size 32 Depth 3	39.50	34.64	30.83	27.87	24.58

Table 4.32. *VVC Mobile SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	0.99	0.98	0.96	0.93	0.84
CTU Size 64 Depth 4	0.99	0.98	0.96	0.93	0.84
CTU Size 64 Depth 3	0.99	0.98	0.96	0.93	0.84
CTU Size 32 Depth 3	0.99	0.98	0.96	0.93	0.84

Table 4.33. *VVC Mobile bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	1805.98	693.52	282.50	136.12	68.85
CTU Size 64 Depth 4	1799.18	691.26	281.84	136.04	68.82
CTU Size 64 Depth 3	1810.23	696.19	283.13	136.49	69.15
CTU Size 32 Depth 3	1819.06	703.80	287.54	139.24	70.76

Mobile video has similar results with other CIF quality videos. Figure 4.19 shows the behavior of Mobile video under different QP levels.

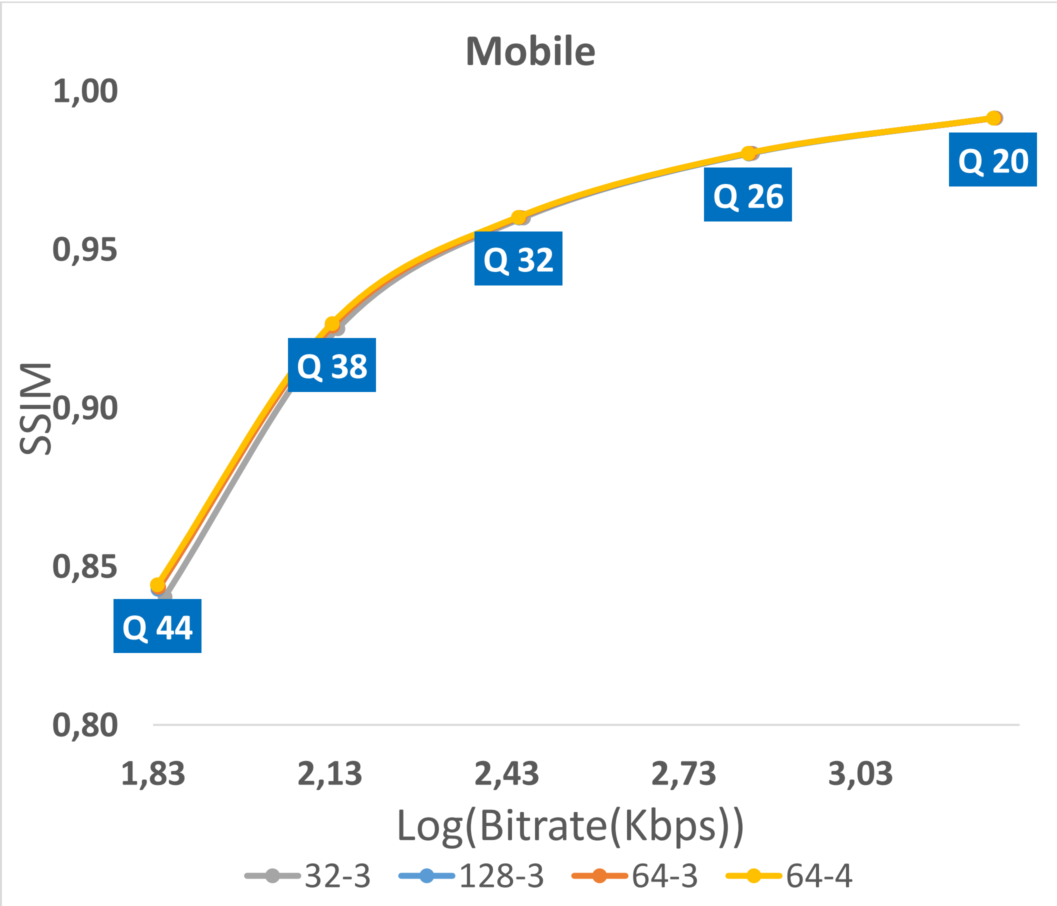

Figure 4.19. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.2.6. VVC Basketball

Table 4.34. *VVC Basketball PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	48.59	47.04	44.51	41.44	37.97
CTU Size 64 Depth 4	47.76	46.54	44.13	41.12	37.65
CTU Size 64 Depth 3	47.73	46.53	44.13	41.12	37.66
CTU Size 32 Depth 3	47.33	46.25	43.82	40.63	37.09

Table 4.35. *VVC Basketball SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	0.99	0.99	0.99	0.98	0.97
CTU Size 64 Depth 4	0.99	0.99	0.99	0.98	0.97
CTU Size 64 Depth 3	0.99	0.99	0.99	0.98	0.97
CTU Size 32 Depth 3	0.99	0.99	0.99	0.98	0.97

Table 4.36. *VVC Basketball bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
CTU Size 128 Depth 3	9345.16	2918.40	1362.22	710.63	370.73
CTU Size 64 Depth 4	9875.37	3072.59	1418.80	739.65	391.11
CTU Size 64 Depth 3	9923.63	3084.48	1421.57	739.33	391.98
CTU Size 32 Depth 3	10825.37	3214.90	1559.84	849.04	483.68

In case of Basketball the results are bit more different than CIF videos for VVC. The efficiency of CTU size 128 depth 3 configuration is more superior than other configurations for all QP levels as it can be evident from Figure 4.20. Table 4.34, Table 4.35 and Table 4.36 shows the numerical results and the difference between CTU 128 depth 3 and other 3 configurations are substantial. This indicates how important of having a higher CTU size for VVC especially for higher quality videos. Since the Basketball video has 3840x2160 dimensions, for encoding stage VVC uses 128x128 more efficiently than in CIF videos. VVC proves how its prepared for higher quality videos by introducing 128x128 CTU size than HEVC.

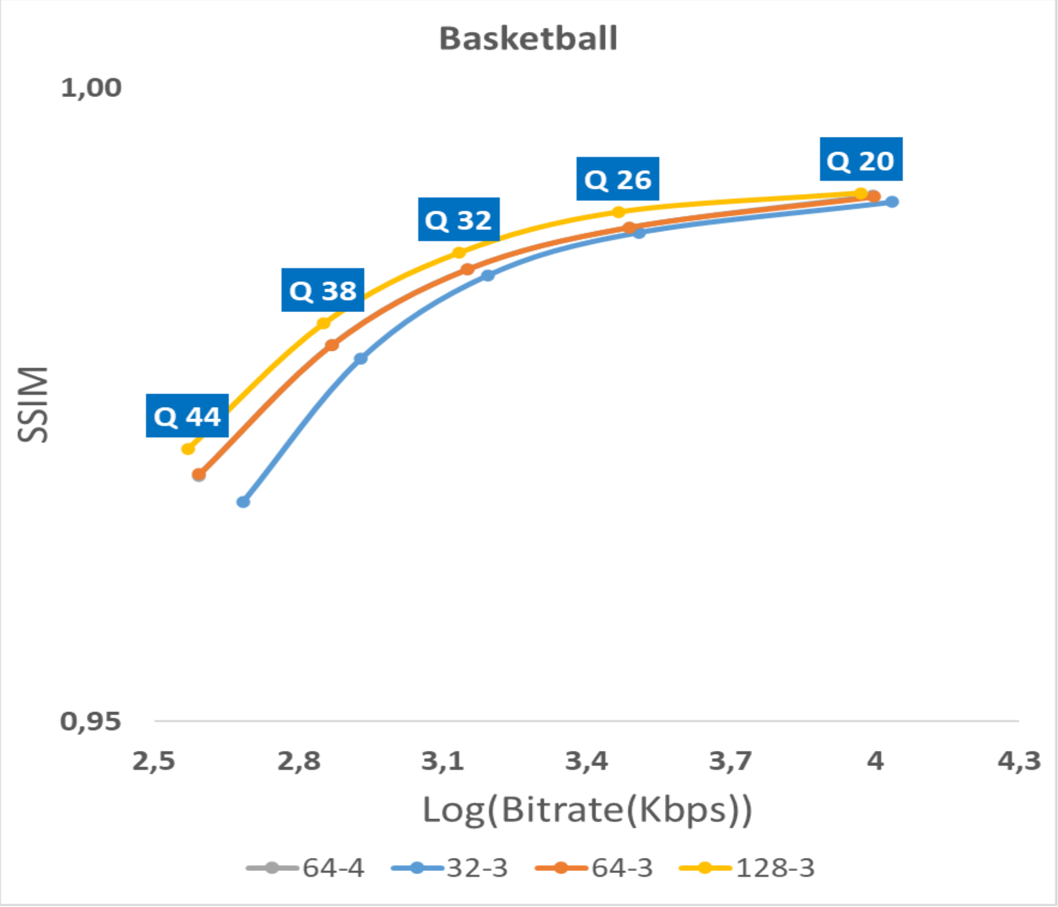

Figure 4.20. Basketball SSIM vs Log(Bitrate(Kbps) graph for different CTU and depth sizes

4.3.2.7. VVC BD Rate results

VVC results are different than HEVC results. CTU 32x32 with depth 3 configuration performs worst for all videos. For HEVC the 32x32 with depth 3 configuration performs worse for only 4K video. However, since the HEVC is an established and rather old codec when compared with VVC, smaller CTU size performance is important, because HEVC needs to consider of lower resolution videos. On the other hand, VVC is a very new and currently being developed codec, it needs to consider higher resolution and complexity videos than HEVC, 4K and 8K or beyond complexities should be the priority quality videos for VVC. Therefore, for these types of videos selecting biggest CTU size is crucial. For CIF videos, 32x32 depth 3 configuration is worse in every case especially for lower complexity videos where using bigger size CTUs improve effectiveness of encoding process.

Moreover, for CIF videos 128x128 depth 3, 64x64 depth 4 and 64x64 depth 3 configurations are perform pretty much same. Since CIF videos have smaller dimensions (352x240) 128x128 CTU size can be unusable and using smaller CU sizes such as 8x8 and 4x4 does not bring high bitrate load to the codecs. Therefore, for CIF videos using higher CTU sizes for VVC does not impactful on quality or bitrate level.

However, when 4K video is processed the difference between 128x128 depth 4 configuration and other configurations are substantial. Figure 4.20 and Figure 4.27 shows the difference and how well 128x128 depth 3 configuration performs. 128x128 depth 3 configuration performs more than %30 better than 32x32 depth 3 configuration in terms of quality and more than %16 better than 64x64 configurations. This performance difference shows how VVC codec is prepared better for higher quality videos by introducing 128x128 CTU size, because using 128x128 configuration for high dimensional videos such as Basketball is important for encoding efficiency.

For VVC codec using the CTU size as big as possible gives the best results, especially for higher quality videos the difference between using maximum CTU size and lower

CTU sizes differs substantially. Using 32x32 CTU size makes no sense for all quality videos, since it performs the worst for all videos. One interesting parameter for VVC is the “HierarchyDepth” parameters. These parameters are introduced while developing the VVC and they are not provided for HEVC. However, changing these depth parameters does not result any significant difference, since for all videos 64x64 depth 4 and 64x64 depth 3 configurations almost result same output. There is very little difference between them. Thus, maybe developers are currently does not take the depth of the partitioning into consideration, because I expected that especially for 4K video the difference should be substantial between 64x64 depth 4 and 64x64 depth 3 configurations.

Video			128-3	64-4	64-3	32-3
Akiyo	128-3	BDBR		0,11	0,57	5,06
		BDPSNR		0,00	-0,02	-0,23
	64-4	BDBR	-0,11		0,46	4,93
		BDPSNR	0,00		-0,02	-0,23
	64-3	BDBR	-0,57	-0,46		4,44
		BDPSNR	0,02	0,02		-0,21
	32-3	BDBR	-4,82	-4,70	-4,26	
		BDPSNR	0,23	0,23	0,21	
			128-3	64-4	64-3	32-3
Bus	128-3	BDBR		-1,08	0,17	1,70
		BDPSNR		0,05	-0,01	-0,07
	64-4	BDBR	1,09		1,26	2,79
		BDPSNR	-0,05		-0,06	-0,12
	64-3	BDBR	-0,17	-1,24		1,53
		BDPSNR	0,01	0,06		-0,07
	32-3	BDBR	-1,67	-2,72	-1,50	
		BDPSNR	0,07	0,12	0,07	
			128-3	64-4	64-3	32-3
Container	128-3	BDBR		1,49	0,33	2,42
		BDPSNR		-0,29	-0,02	-0,10
	64-4	BDBR	-1,47		-1,16	0,90
		BDPSNR	0,29		0,28	0,14
	64-3	BDBR	-0,33	1,18		2,09
		BDPSNR	0,02	-0,28		-0,09
	32-3	BDBR	-2,36	-0,89	-2,05	
		BDPSNR	0,10	-0,14	0,09	
			128-3	64-4	64-3	32-3
Foreman	128-3	BDBR		-0,50	0,16	6,60
		BDPSNR		0,02	-0,01	-0,26
	64-4	BDBR	0,50		0,67	7,13
		BDPSNR	-0,02		-0,03	-0,28
	64-3	BDBR	-0,16	-0,66		6,42
		BDPSNR	0,01	0,03		-0,25
	32-3	BDBR	-6,19	-6,66	-6,04	
		BDPSNR	0,26	0,28	0,25	
			128-3	64-4	64-3	32-3
Mobile	128-3	BDBR		-0,97	-0,32	1,38
		BDPSNR		0,04	0,01	-0,06
	64-4	BDBR	0,98		0,66	2,37
		BDPSNR	-0,04		-0,03	-0,11
	64-3	BDBR	0,32	-0,66		1,70
		BDPSNR	-0,01	0,03		-0,08
	32-3	BDBR	-1,36	-2,31	-1,67	
		BDPSNR	0,06	0,11	0,08	
			128-3	64-4	64-3	32-3
Basketball	128-3	BDBR		19,33	19,83	46,96
		BDPSNR		-0,62	-0,64	-1,34
	64-4	BDBR	-16,20		0,68	26,60
		BDPSNR	0,62		-0,02	-0,72
	64-3	BDBR	-16,55	-0,68		25,97
		BDPSNR	0,64	0,02		-0,70
	32-3	BDBR	-31,95	-21,01	-20,62	
		BDPSNR	1,34	0,72	0,70	

Figure 4.21. VVC BD Rates of all videos for different CTU and depth sizes

Figure 4.22. Akiyo BD Rates for VVC

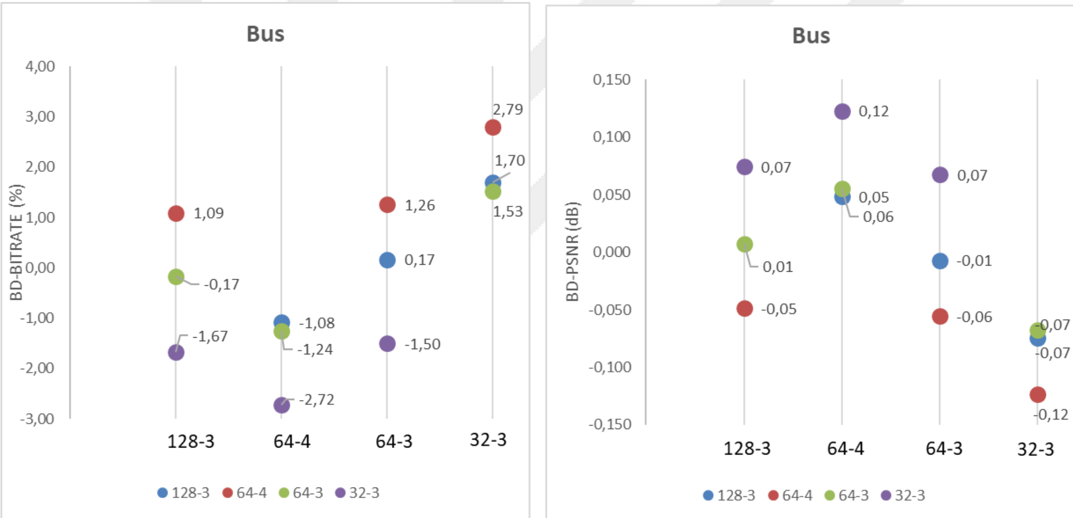

Figure 4.23. Bus BD Rates for VVC

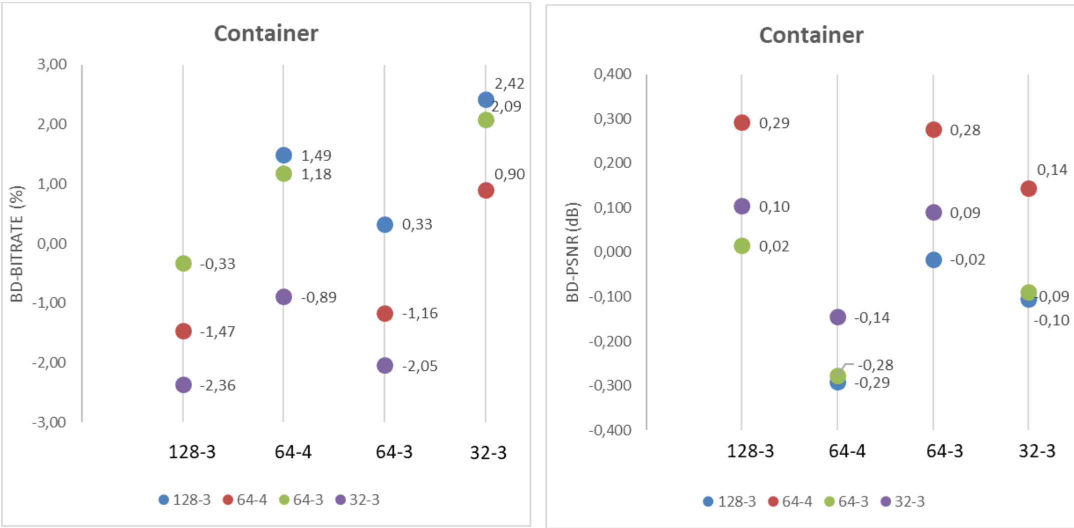

Figure 4.24. Container BD Rates for VVC

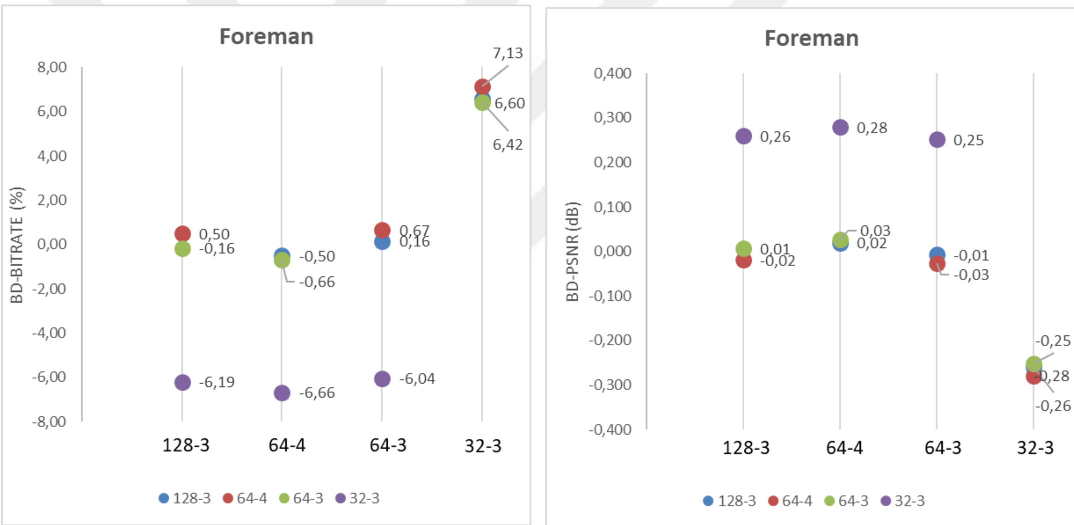

Figure 4.25. Foreman BD Rates for VVC

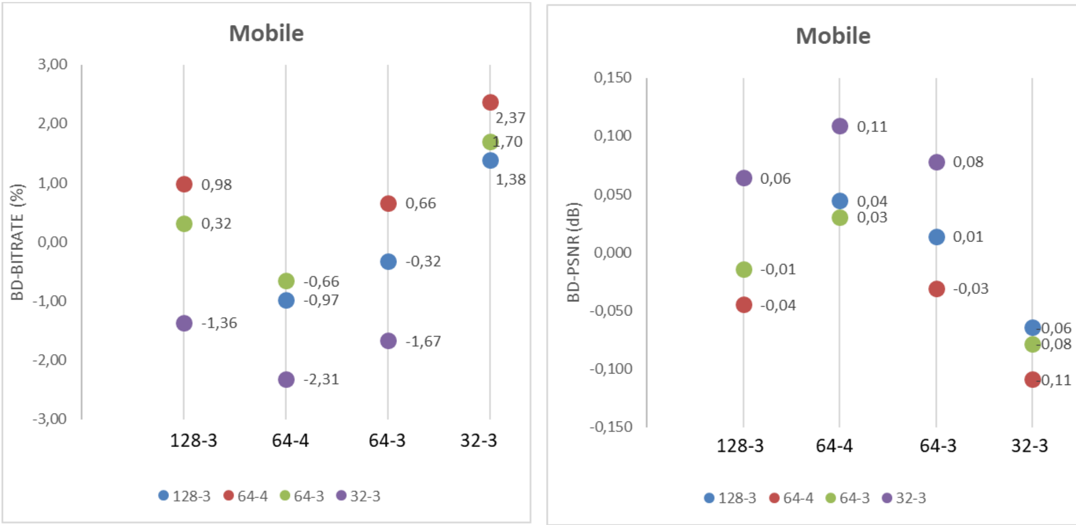

Figure 4.26. Mobile BD Rates for VVC

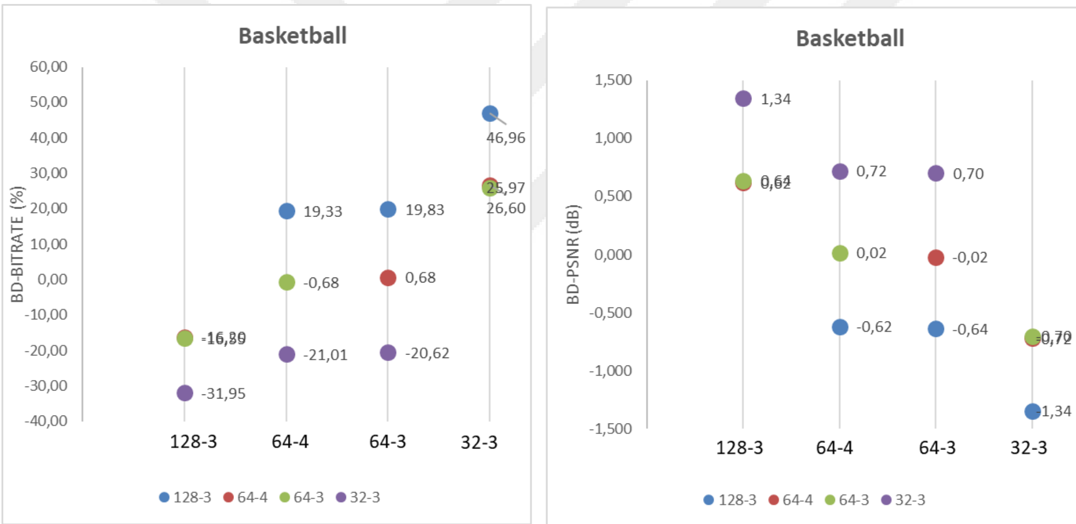

Figure 4.27. Basketball BD Rates for VVC

4.3.3. AV1 CTU Sizes

In the configuration file “--min-partition-size” and “--max-partition-size” parameters are adjusted such that the depth is acquired in a similar way with HEVC. When “--min-partition-size” is set to 8 and “--max-partition-size” is set to 64 than partition depth is satisfied as 4. “--max-partition-size” is set to 128, 64 and 32. For all 3 cases “min-partition-size” is set to 8 and for the case when ““‘--max-partition-size” is 64, “min-partition-size” is also set to 16. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8. Quantization parameter is set to 20, 26, 32, 38, 44 while performing the tests. GF interval is fixed to 8. Rate control is off.

4.3.3.1. AV1 Akiyo

Table 4.37. *AV1 Akiyo PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	48.40	47.12	45.75	44.58	43.37
Min Part Size 8 Max Part Size 32	48.39	47.10	45.74	44.56	43.33
Min Part Size 16 Max Part Size 64	48.07	46.88	45.57	44.39	43.18
Min Part Size 8 Max Part Size 128	47.32	46.11	44.60	43.47	42.30

Table 4.38. *AV1 Akiyo SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	0.99	0.99	0.99	0.99	0.98
Min Part Size 8 Max Part Size 32	0.99	0.99	0.99	0.99	0.98
Min Part Size 16 Max Part Size 64	0.99	0.99	0.99	0.99	0.98
Min Part Size 8 Max Part Size 128	0.99	0.99	0.99	0.98	0.98

Table 4.39. *AV1 Akiyo bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	339.42	248.45	175.63	132.72	97.81
Min Part Size 8 Max Part Size 32	339.36	248.83	176.35	133.23	98.34
Min Part Size 16 Max Part Size 64	345.51	257.24	184.48	139.00	102.63
Min Part Size 8 Max Part Size 128	257.98	191.70	132.68	99.40	75.32

AV1 uses different configuration inputs than HEVC and VVC. It takes inputs to specify maximum and minimum partition sizes, and by defining maximum and minimum sizes the depth of partitioning is also automatically selected. In the Figure 4.28 the notations are used to make it similar to HEVC and VVC. In AV1 for lower complexity videos such as Akiyo performance of CTU size 128 depth 5 configuration is worse than others. Since the codec is specified to be able to use 128x128 CTU sizes both bitrate and quality values are decreasing for video Akiyo. Table 4.37, Table 4.38 and Table 4.39 are supporting this and the CTU 128 depth 5 configuration outputs more than 1 dB PSNR lower than the CTU 64 depth 4 configuration and around 80 kBps less bitrate.

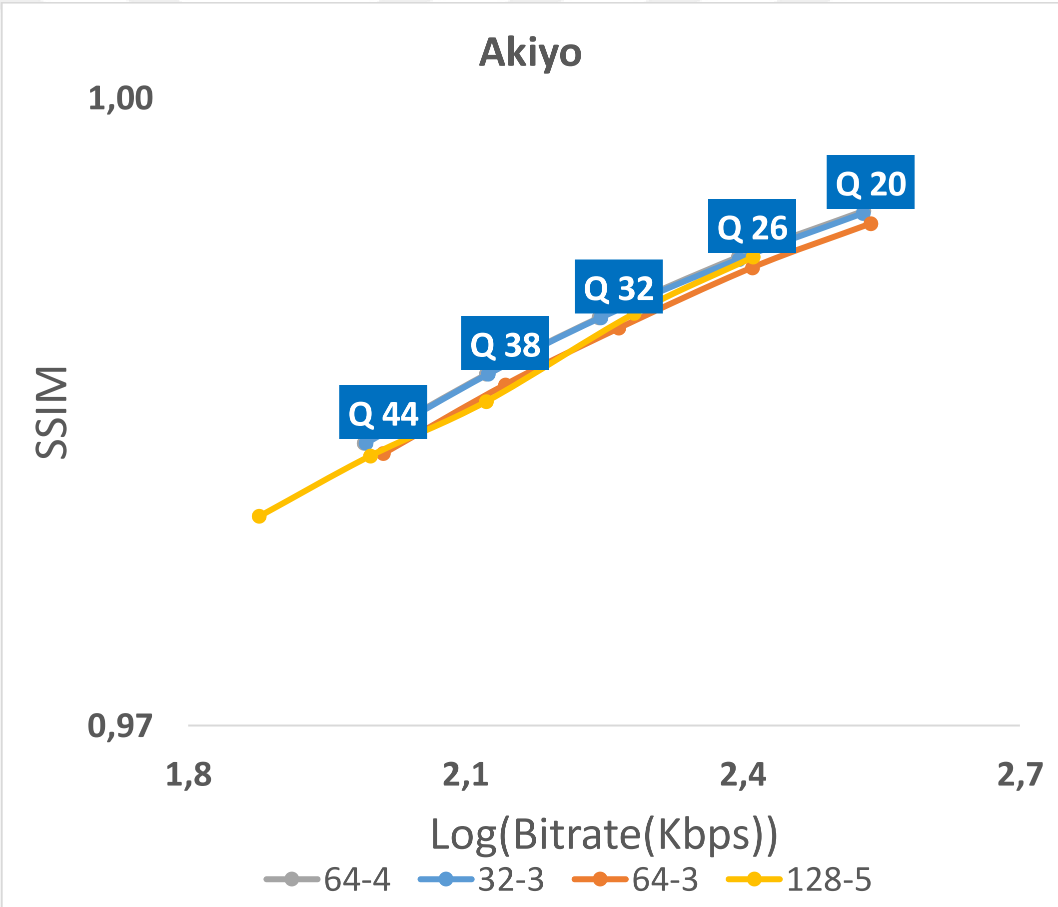

Figure 4.28. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.3.2. AV1 Bus

Table 4.40. *AV1 Bus PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	41.60	40.12	38.37	36.68	34.93
Min Part Size 8 Max Part Size 32	41.60	40.12	38.38	36.68	34.93
Min Part Size 16 Max Part Size 64	41.41	39.88	38.10	36.36	34.65
Min Part Size 8 Max Part Size 128	41.27	39.87	38.11	36.41	34.65

Table 4.41. *AV1 Bus SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	0.99	0.99	0.98	0.97	0.96
Min Part Size 8 Max Part Size 32	0.99	0.99	0.98	0.97	0.96
Min Part Size 16 Max Part Size 64	0.99	0.98	0.98	0.97	0.96
Min Part Size 8 Max Part Size 128	0.99	0.98	0.98	0.97	0.96

Table 4.42. *AV1 Bus bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	1968.83	1502.40	1092.19	793.04	572.85
Min Part Size 8 Max Part Size 32	1968.30	1502.54	1092.53	794.89	573.85
Min Part Size 16 Max Part Size 64	2136.40	1636.37	1198.56	868.69	622.43
Min Part Size 8 Max Part Size 128	1795.44	1380.39	995.83	717.78	515.40

For video Bus the performance difference between configurations is more severe. Figure 4.29 shows the behavior of the configurations. Again CTU 128 depth 5 configuration has lowest bitrate values; however it has better PSNR results than CTU 64 depth 3 configuration as Table 4.40, Table 4.41 and Table 4.42 supports. The configurations CTU 64 depth 4 and CTU 32 depth 3 configurations results almost same. In Bus video CTU 64 depth 3 configuration performs worse than others because the video has most spatial behavior and for more complex videos using smallest possible CU sizes is more important. The only configuration that can not partition into 4x4 CU size is CTU size 64 depth 3 configuration; therefore, its performance is the worst for Bus video.

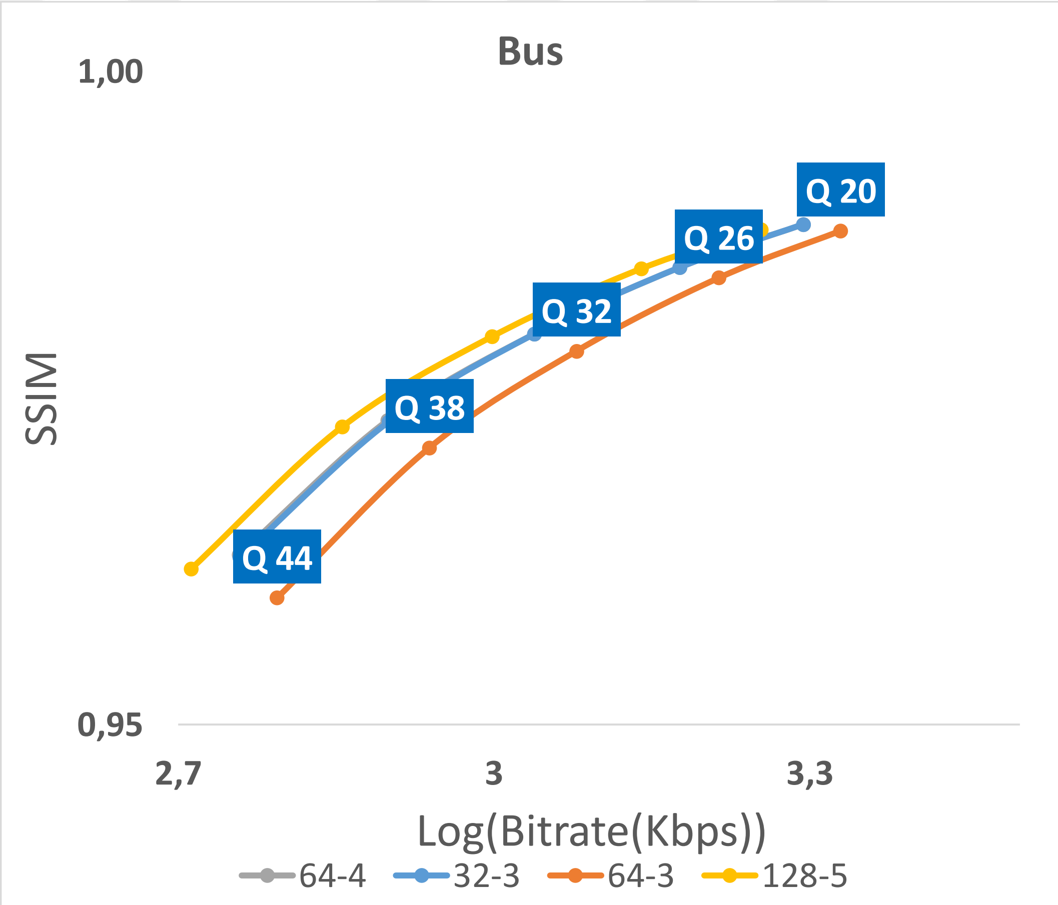

Figure 4.29. Bus SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.3.3. AV1 Container

Table 4.43. *AV1 Container PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	43.88	42.74	41.49	40.23	39.11
Min Part Size 8 Max Part Size 32	43.89	42.75	41.48	40.22	39.09
Min Part Size 16 Max Part Size 64	43.79	42.64	41.39	40.12	39.00
Min Part Size 8 Max Part Size 128	43.59	42.56	41.32	40.05	38.82

Table 4.44. *AV1 Container SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	0.98	0.98	0.97	0.96	0.95
Min Part Size 8 Max Part Size 32	0.98	0.98	0.97	0.96	0.95
Min Part Size 16 Max Part Size 64	0.98	0.98	0.97	0.96	0.95
Min Part Size 8 Max Part Size 128	0.98	0.98	0.97	0.96	0.95

Table 4.45. *AV1 Container bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	601.57	443.20	321.34	236.33	179.39
Min Part Size 8 Max Part Size 32	603.95	445.98	322.07	237.33	179.99
Min Part Size 16 Max Part Size 64	617.16	455.99	330.52	243.07	184.48
Min Part Size 8 Max Part Size 128	564.72	429.30	310.72	225.56	166.54

Container video has smaller SI value like Akiyo, thus the results become close to each other with Akiyo video as in Figure 4.30. CTU 128 depth 5 has smallest bitrate as well as quality outputs for every QP level. Table 4.43, Table 4.44 and Table 4.45 shows that again configurations CTU 64 depth 4 and CTU 32 depth 3 configurations are close to each other.

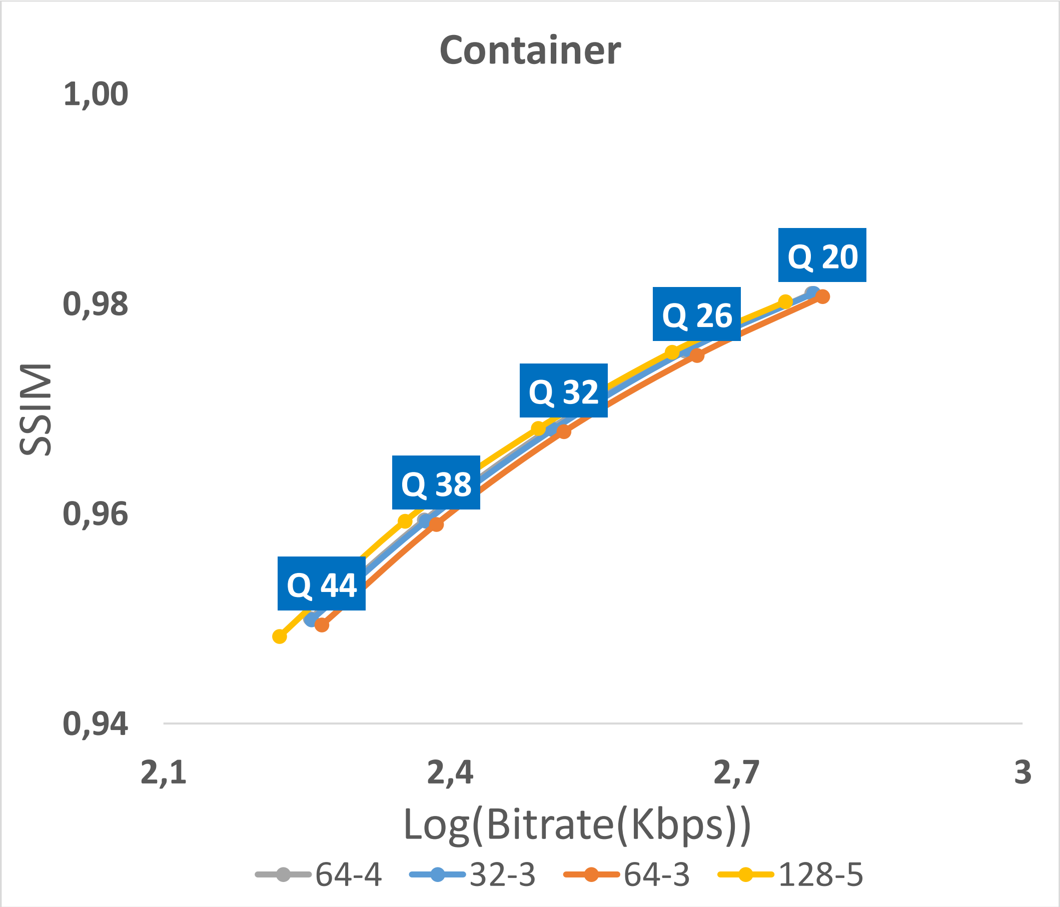

Figure 4.30. Container SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.3.4. AV1 Foreman

Table 4.46. *AV1 Foreman PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	42.92	41.89	40.69	39.54	38.34
Min Part Size 8 Max Part Size 32	42.92	41.89	40.68	39.53	38.31
Min Part Size 16 Max Part Size 64	42.70	41.64	40.42	39.26	38.06
Min Part Size 8 Max Part Size 128	42.62	41.58	40.29	39.04	37.69

Table 4.47. *AV1 Foreman SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	0.98	0.98	0.97	0.97	0.96
Min Part Size 8 Max Part Size 32	0.98	0.98	0.97	0.97	0.96
Min Part Size 16 Max Part Size 64	0.98	0.98	0.97	0.97	0.96
Min Part Size 8 Max Part Size 128	0.98	0.98	0.97	0.96	0.95

Table 4.48. *AV1 Foreman bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	848.01	638.58	455.63	334.94	245.03
Min Part Size 8 Max Part Size 32	847.63	639.01	457.39	337.06	246.36
Min Part Size 16 Max Part Size 64	927.35	694.02	494.32	360.30	261.09
Min Part Size 8 Max Part Size 128	752.70	561.19	399.19	287.89	205.29

Foreman Video shows similar results like other CIF videos. CTU 128 depth 5 configuration results lower PSNR as well as lower bitrate while CTU 64 depth 4 and CTU 32 depth 3 configurations result almost same as it can be seen from Figure 4.31. From Table 4.47, Table 4.46 and Table 4.48 the data shows that CTU 64 depth 4 and CTU 32 depth 3 configurations result better PSNR as well as lower bitrate than CTU 64 depth 3 configuration, meaning their overall efficiency better than CTU 64 depth 3 configuration.

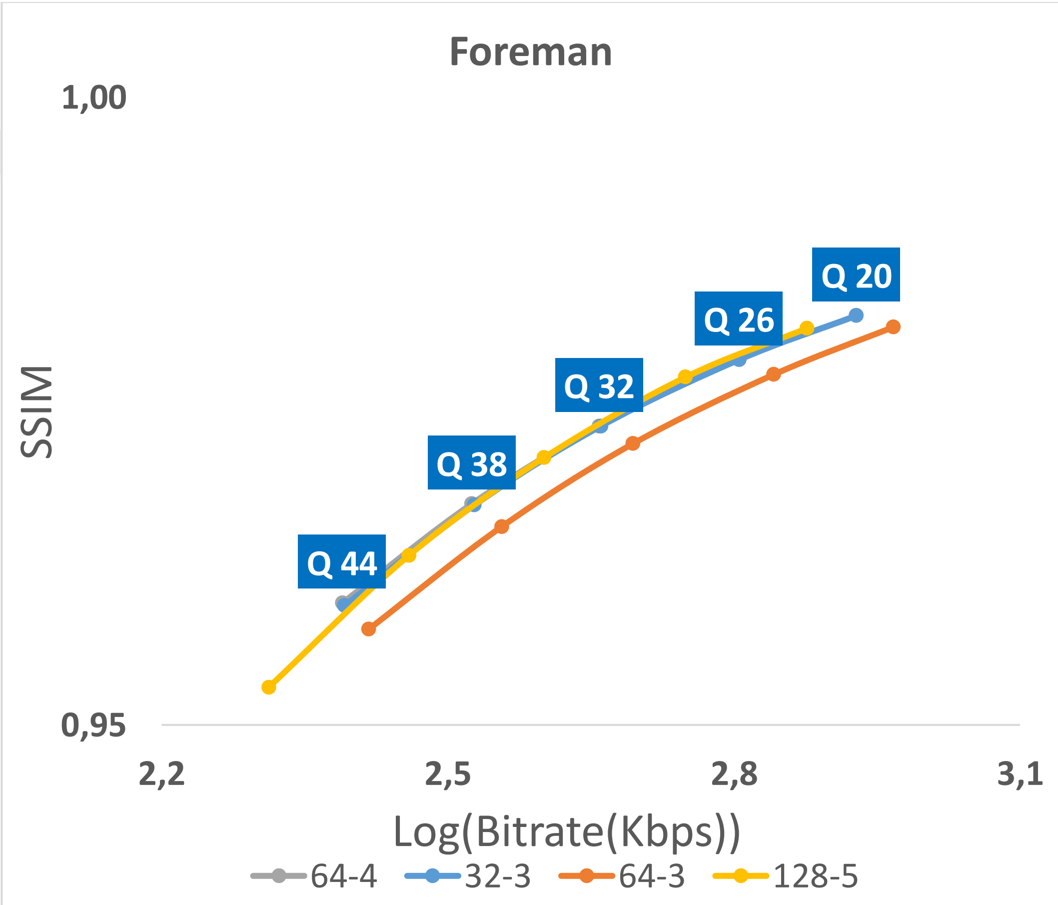

Figure 4.31. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.3.5. AV1 Mobile

Table 4.49. *AV1 Mobile PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	41.34	39.53	37.62	35.81	34.21
Min Part Size 8 Max Part Size 32	41.34	39.53	37.62	35.81	34.21
Min Part Size 16 Max Part Size 64	41.16	39.31	37.37	35.53	33.91
Min Part Size 8 Max Part Size 128	40.73	38.98	36.98	35.25	33.63

Table 4.50. *AV1 Mobile SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	0.99	0.99	0.99	0.98	0.98
Min Part Size 8 Max Part Size 32	0.99	0.99	0.99	0.98	0.98
Min Part Size 16 Max Part Size 64	0.99	0.99	0.99	0.98	0.98
Min Part Size 8 Max Part Size 128	0.99	0.99	0.99	0.98	0.98

Table 4.51. *AV1 Mobile bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	2635.47	1989.62	1423.28	1009.96	725.48
Min Part Size 8 Max Part Size 32	2635.43	1991.97	1422.99	1008.93	725.99
Min Part Size 16 Max Part Size 64	2824.28	2141.09	1532.95	1085.93	774.71
Min Part Size 8 Max Part Size 128	2362.24	1764.34	1227.94	864.06	609.59

For video Mobile the results have similar behavior as other CIF videos. CTU 128 depth 5 configuration has lowest bitrate values as well as lowest quality values for all QP levels, while configurations CTU 64 depth 4 and CTU 32 depth 3 configurations perform almost even.

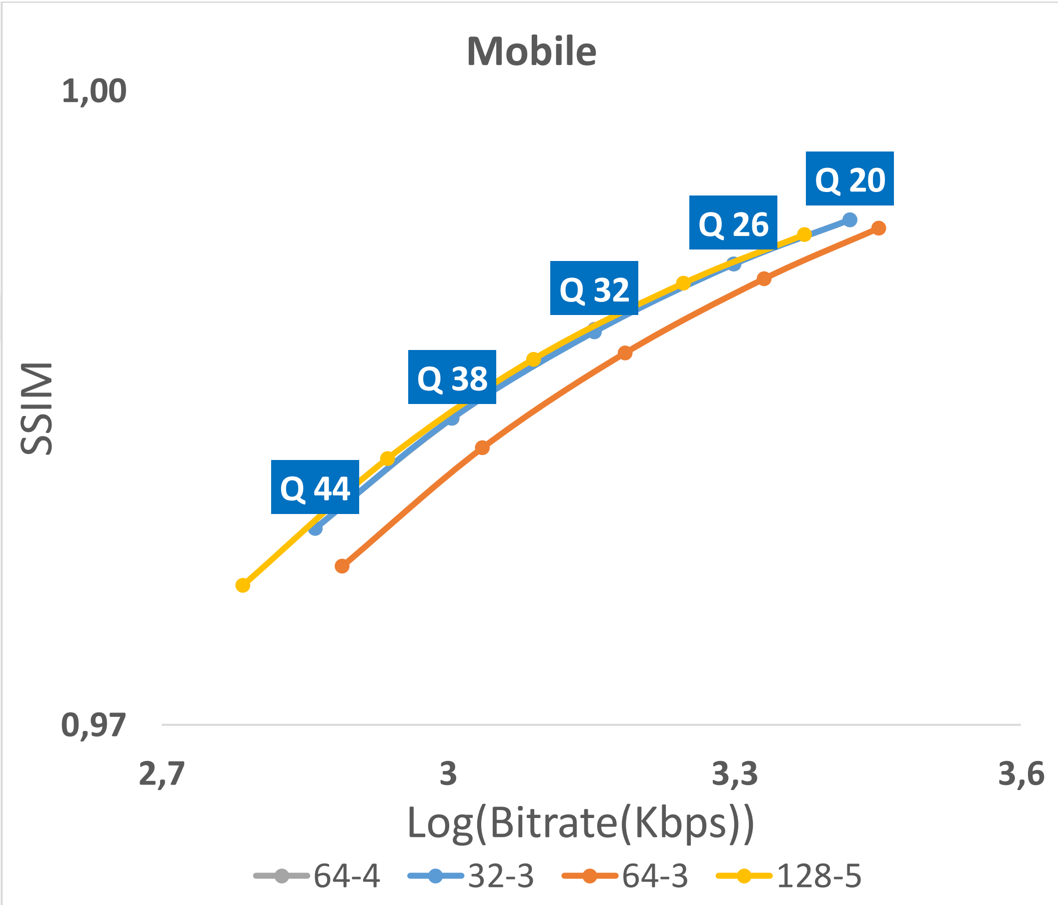

Figure 4.32. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.3.3.6. AV1 Basketball

Table 4.52. *AV1 Basketball PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	48.27	47.83	47.24	46.53	45.64
Min Part Size 8 Max Part Size 32	48.26	47.83	47.20	46.48	45.57
Min Part Size 16 Max Part Size 64	48.22	47.79	47.20	46.49	45.60
Min Part Size 8 Max Part Size 128	48.12	47.68	47.02	46.26	45.38

Table 4.53. *AV1 Basketball SSIM values (dB) with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
Min Part Size 8 Max Part Size 64	0.99	0.99	0.99	0.99	0.99
Min Part Size 8 Max Part Size 32	0.99	0.99	0.99	0.99	0.99
Min Part Size 16 Max Part Size 64	0.99	0.99	0.99	0.99	0.99
Min Part Size 8 Max Part Size 128	0.99	0.99	0.99	0.99	0.99

Table 4.54. *AV1 Basketball bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP =26	QP =32	QP =38	QP =44
Min Part Size 8 Max Part Size 64	10483.71	7014.53	4775.31	3421.87	2481.27
Min Part Size 8 Max Part Size 32	10724.28	7238.06	4968.68	3587.95	2623.53
Min Part Size 16 Max Part Size 64	10582.05	7131.20	4848.17	3461.16	2497.16
Min Part Size 8 Max Part Size 128	8739.60	6079.05	4259.23	3116.26	2335.97

In Basketball Video, the results are different from CIF quality videos. The biggest difference between CIF videos and 4K quality video, as in the case of VVC having bigger size CTU sizes is more important, since the dimensions are large frequency of using bigger CU sizes is increased. Figure 4.33 shows how the gap between CTU size 64 depth 3 configuration and other configurations disappears. This implies having a large size of CTU is more important than partitioning into smallest CU size possible which is 4x4 for AV1. Therefore, like in VVC, CTU 64 depth 3 performance better for higher quality videos, while it performs worse than others for lower quality videos since smaller CU sizes are important for lower quality videos. Table 4.52, Table 4.53 and Table 4.54 illustrates numerical values and CTU 128 depth 5 configuration lacks behind similar to CIF videos in both bitrate and quality for all QP levels.

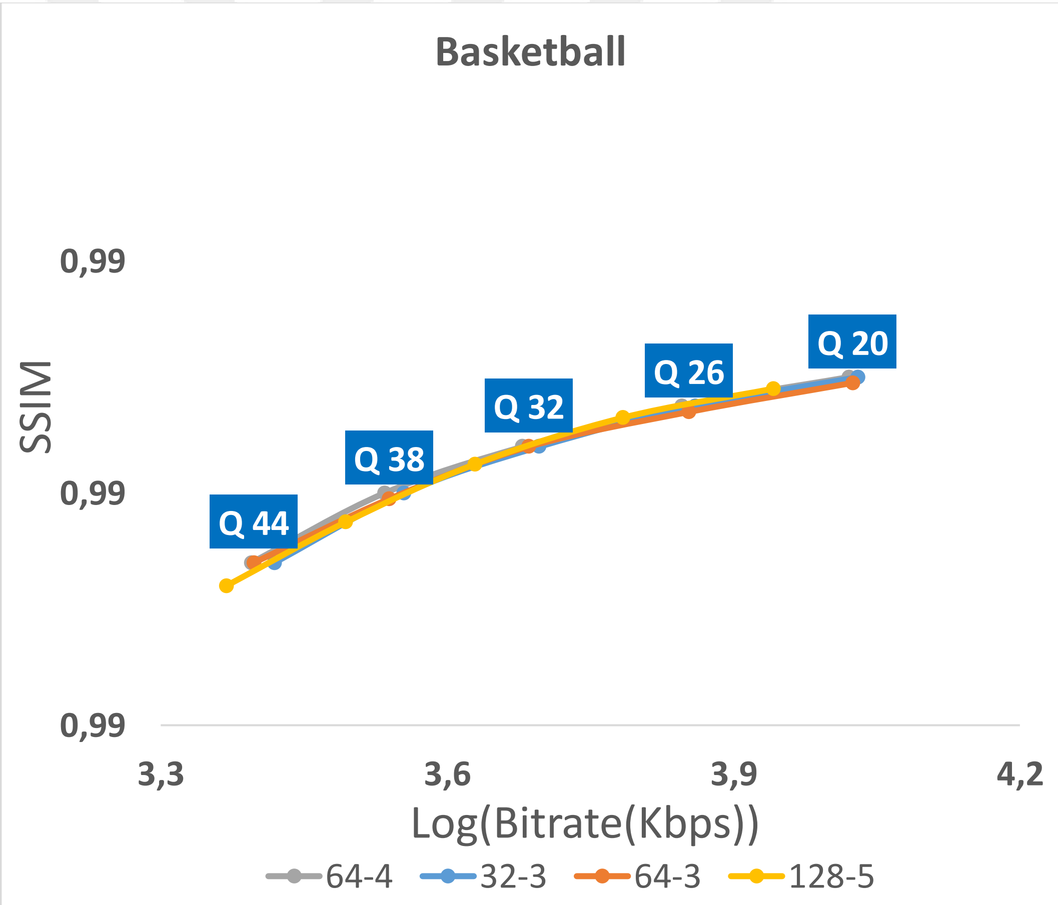

Figure 4.33. Basketball SSIM vs Log(Bitrate(Kbps) graph for different CTU and depth sizes

4.3.3.7. AV1 BD Rate results

AV1 results show similarity to VVC. For CIF videos going to the smallest possible CU size is the most important aspect of encoding process. Figure 4.34 shows the BD rate results and for CIF quality videos the worst results acquired when the "—min-partition-size" parameter is set to 16. This means that the configuration is 64x64 depth 3 to be at the same page with VVC and HEVC. Best results always achieved when 128x128 depth 5 configuration is selected. Since having the widest possibilities for CTU sizes is acquired from 128x128 depth 5 configuration. By this configuration codec can use CU sizes from 128x128 to 4x4. Overall, for CIF videos 64x64 depth 4 and 32x32 depth 3 configurations show close results, while 128x128 depth 5 has best results, these all three configurations are going to the same smallest size possible CU size which is 4x4. 64x64 depth 3 configuration lacks behind other three configurations significantly.

When 4K video is considered, the results are different from CIF video results. For all configurations, the results are near to each other. 128x128 depth 5 configuration has the best performance again, since it has the widest possible CTU size selections. The most important point is that 64x64 depth 3 configuration, which by far the worst one for CIF videos, performs better than the configuration 32x32 depth 3. Since the dimensions of the 4K video is large, AV1 codec does not partition into the smallest possible CU as in the case for CIF videos. Therefore, choosing a bigger CTU size rather than going to the smallest CU size is more important for higher resolution videos. Therefore, the configuration 64x64 depth 3 is better than 32x32 depth 3 configuration. This result can be deducted from the difference between 64x64 depth 4 and 32x32 depth 3 configurations. For CIF videos these 2 results are almost same for all type of videos; however, for 4K video 64x64 depth 4 configuration save %5,6 bitrate at the same quality level and results 0,11 dB better PSNR at the same bitrate level.

The other important point for AV1 is that the difference between 128x128 depth 5 and 64x64 depth 4 configuration results. There is only %0,43 bitrate efficiency between two configurations at the same quality level. This number is very small, I would expect much steeper difference since 128x128 CTU size is a new concept and it should have serious advantage over other configurations especially for 4K video, like in VVC codec. This result tells that AV1 does not use the biggest CTU size that often in encoding process.

Video						
Akiyo			128-5	64-4	64-3	32-3
	128-5	BDBR		0,55	10,42	1,36
		BDPSNR		-0,02	-0,40	-0,05
	64-4	BDBR	-0,55		9,91	0,73
		BDPSNR	0,02		-0,38	-0,03
	64-3	BDBR	-9,44	-9,01		-8,32
		BDPSNR	0,40	0,38		0,35
	32-3	BDBR	-1,34	-0,72	9,08	
		BDPSNR	0,05	0,03	-0,35	
Bus			128-3	64-4	64-3	32-3
	128-3	BDBR		4,48	20,42	4,59
		BDPSNR		-0,23	-1,01	-0,24
	64-4	BDBR	-4,28		15,09	0,10
		BDPSNR	0,23		-0,77	-0,01
	64-3	BDBR	-16,96	-13,11		-13,02
		BDPSNR	1,01	0,77		0,77
	32-3	BDBR	-4,39	-0,10	14,97	
		BDPSNR	0,24	0,01	-0,77	
Container			128-3	64-4	64-3	32-3
	128-3	BDBR		-0,71	4,87	-0,08
		BDPSNR		0,03	-0,19	0,00
	64-4	BDBR	0,71		5,57	0,61
		BDPSNR	-0,03		-0,21	-0,02
	64-3	BDBR	-4,64	-5,28		-4,70
		BDPSNR	0,19	0,21		0,19
	32-3	BDBR	0,08	-0,60	4,93	
		BDPSNR	0,00	0,02	-0,19	
Foreman			128-3	64-4	64-3	32-3
	128-3	BDBR		2,95	19,42	3,54
		BDPSNR		-0,11	-0,66	-0,13
	64-4	BDBR	-2,86		16,20	0,54
		BDPSNR	0,11		-0,55	-0,02
	64-3	BDBR	-16,26	-13,94		-13,44
		BDPSNR	0,66	0,55		0,53
	32-3	BDBR	-3,42	-0,54	15,53	
		BDPSNR	0,13	0,02	-0,53	
Mobile			128-3	64-4	64-3	32-3
	128-3	BDBR		3,28	16,32	3,28
		BDPSNR		-0,17	-0,82	-0,17
	64-4	BDBR	-3,18		12,35	0,00
		BDPSNR	0,17		-0,65	0,00
	64-3	BDBR	-14,03	-11,00		-11,00
		BDPSNR	0,82	0,65		0,65
	32-3	BDBR	-3,17	0,00	12,36	
		BDPSNR	0,17	0,00	-0,65	
Basketball			128-3	64-4	64-3	32-3
	128-3	BDBR		0,43	3,64	6,55
		BDPSNR		-0,01	-0,07	-0,13
	64-4	BDBR	-0,43		3,35	5,93
		BDPSNR	0,01		-0,06	-0,11
	64-3	BDBR	-3,52	-3,24		2,60
		BDPSNR	0,07	0,06		-0,04
	32-3	BDBR	-6,15	-5,60	-2,53	
		BDPSNR	0,13	0,11	0,04	

Figure 4.34. AV1 BD Rates of all videos for different CTU and depth sizes

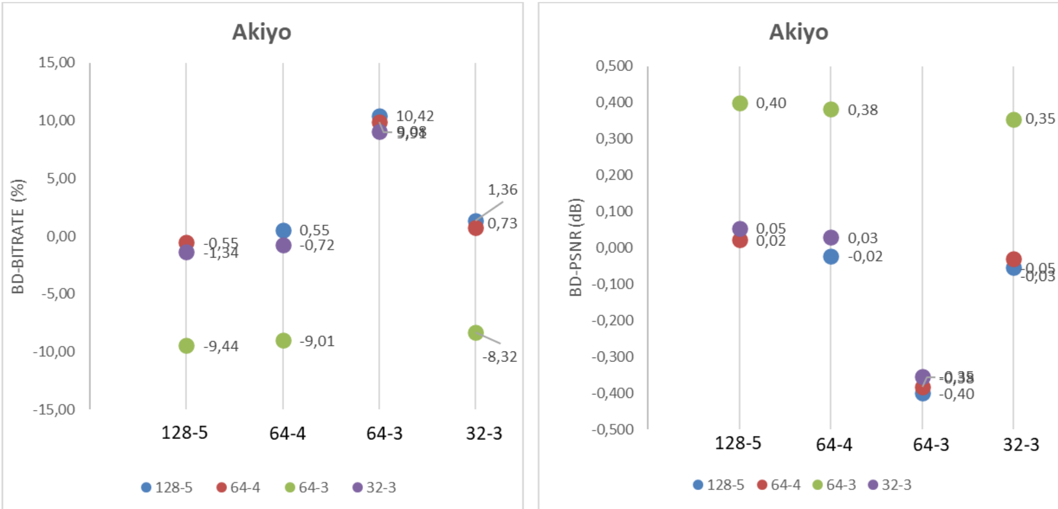

Figure 4.35. Akiyo BD Rates for AV1

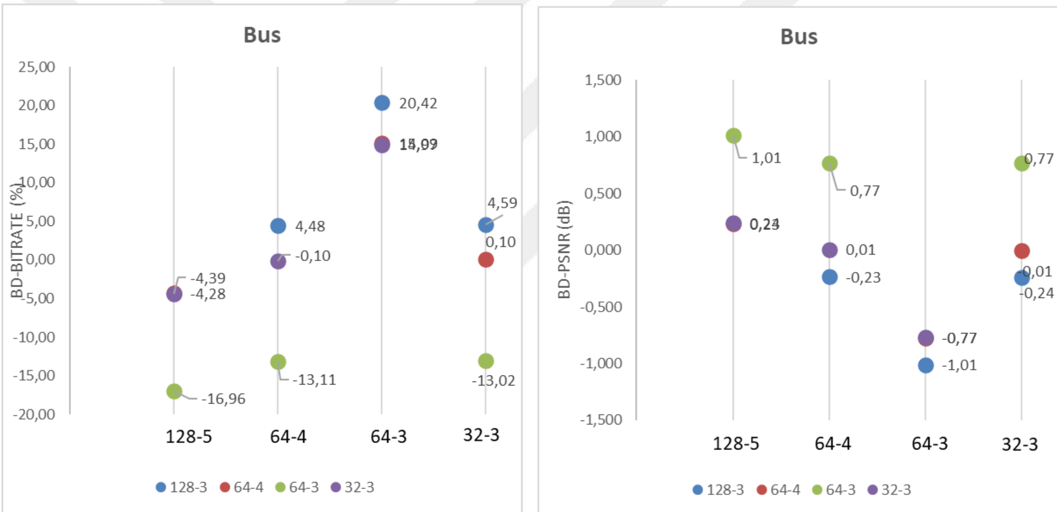

Figure 4.36. Bus BD Rates for AV1

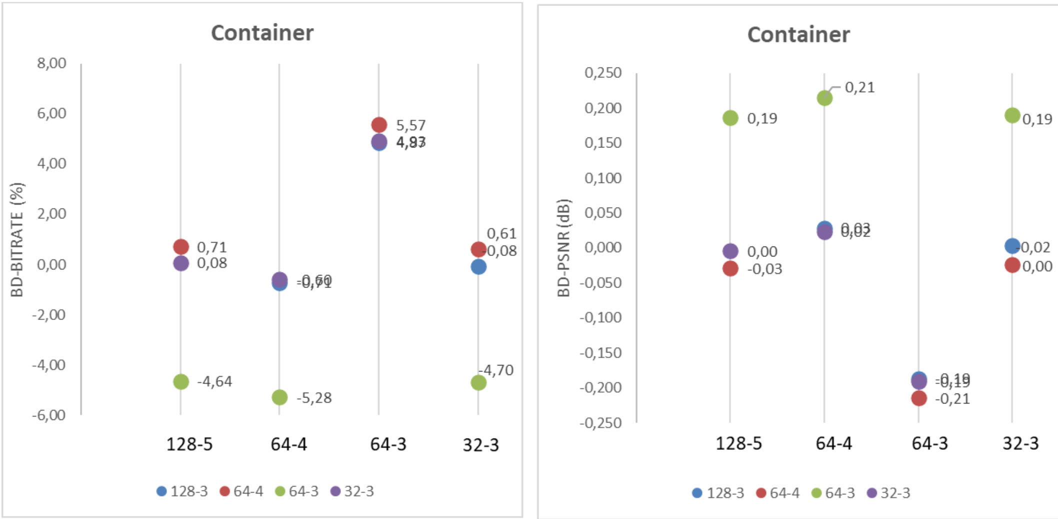

Figure 4.37. Container BD Rates for AV1

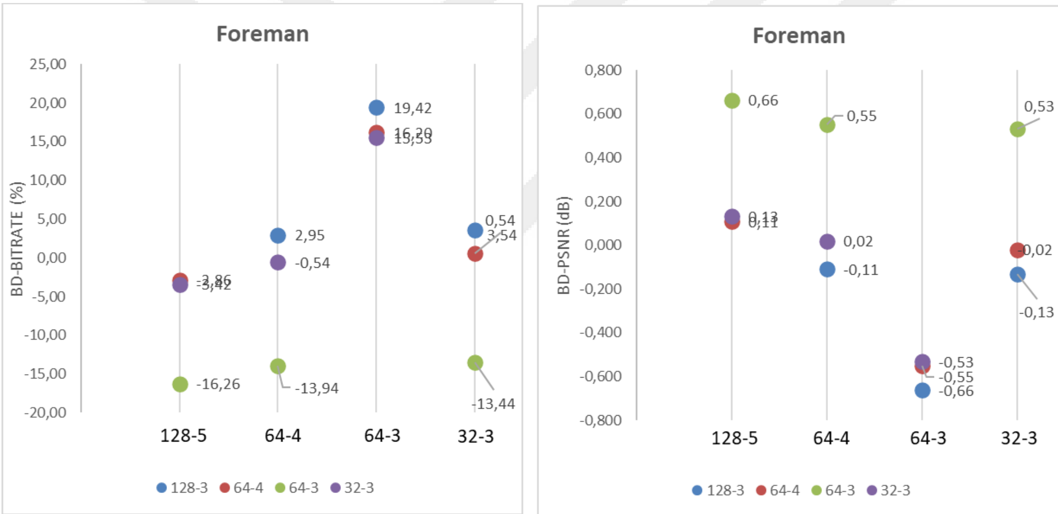

Figure 4.38. Foreman BD Rates for AV1

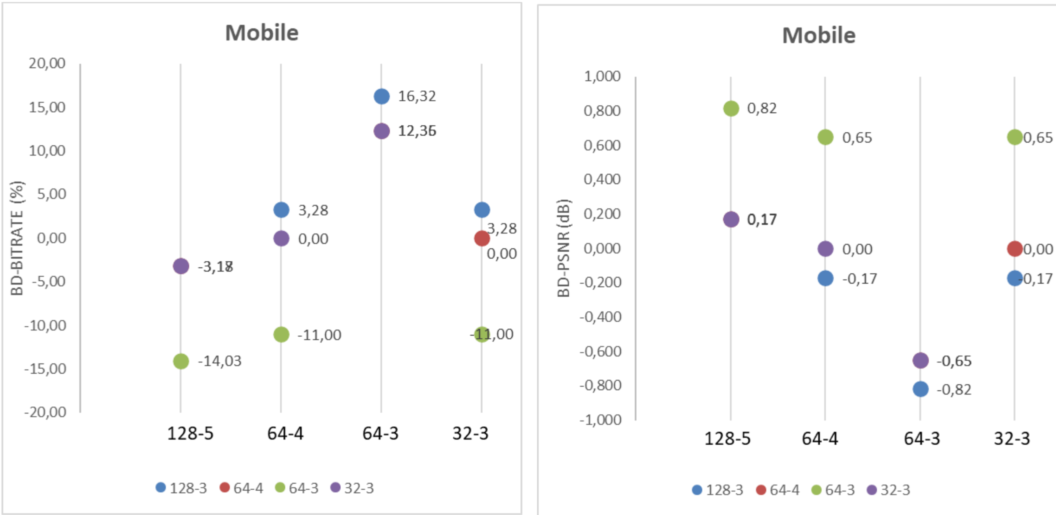

Figure 4.39. Mobile BD Rates for AV1

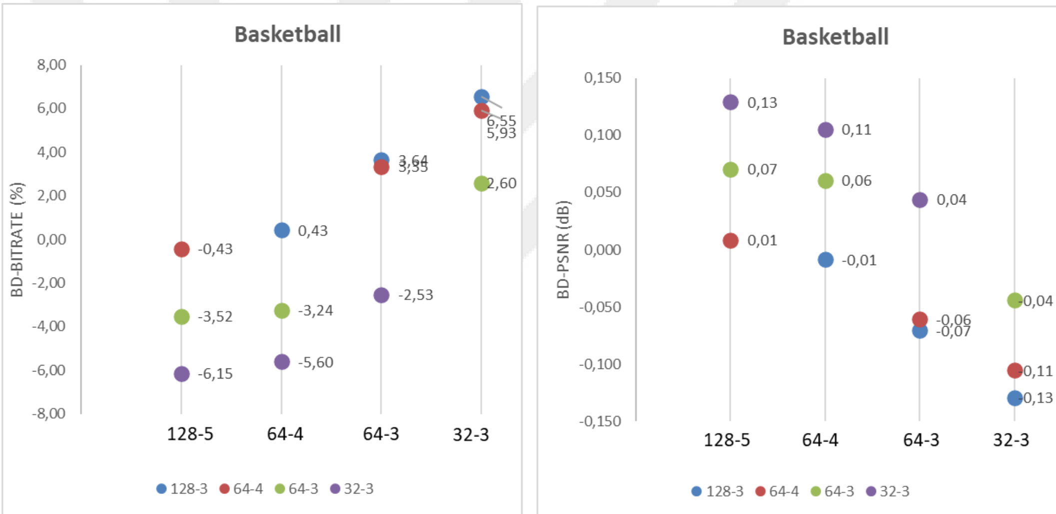

Figure 4.40. *Basketball BD Rates for AV1*

4.4. I Frame Difference

4.4.1. HEVC I Frame Difference

In the configuration file “IntraPeriod” parameters is adjusted at 16, 32, 64 and 128 while performing I frame difference test. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8. Quantization parameter is set to 20, 26, 32, 38, 44 while performing the tests. Maximum CU width and height are adjusted at 64, and partition depth is fixed at 3. Rate control is off. Other parameters are all taken as default parameters. HEVC codec uses all B frames for encoding process.

4.4.1.1. HEVC Akiyo

Table 4.55. *HEVC Akiyo PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	45.35	42.69	39.55	36.07	32.28
I Frame 32	45.12	42.30	39.09	35.64	31.92
I Frame 64	45.01	42.13	38.90	35.45	31.80
I Frame 128	44.92	41.97	38.68	35.21	31.65

Table 4.56. *HEVC Akiyo SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.98	0.97	0.95	0.90
I Frame 32	0.99	0.98	0.97	0.94	0.90
I Frame 64	0.99	0.98	0.97	0.94	0.90
I Frame 128	0.99	0.98	0.97	0.94	0.89

Table 4.57. *HEVC Akiyo bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	266.18	135.61	73.76	42.15	24.61
I Frame 32	195.07	91.22	46.64	25.99	15.17
I Frame 64	164.68	72.98	35.72	19.38	11.41
I Frame 128	150.54	63.84	30.02	15.95	9.51

In HEVC changing I frame period results almost same for all videos. In Akiyo video Figure 4.41 illustrates the different behaviors of the I frame configurations. Increasing the I frame interval decreases the bitrate as well as the quality of the video. There is a tradeoff for I frame interval change. Since an intra frame is a frame that is fully encoded by only intra predictions and no inter predictions, it consists more data than inter frames. This results that codec using more bitrate for encoding process. On the other hand, inter predictions will be more accurate with using more frequent I frames, since there is more frequent I frames to perform accurate inter predictions. Thus, the results are expected, and they are accurate. Table 4.55, Table 4.56 and Table 4.57 supports the graph and it can be seen that minimum PSNR and bitrate values for all QP values are acquired for I frame 128 configuration.

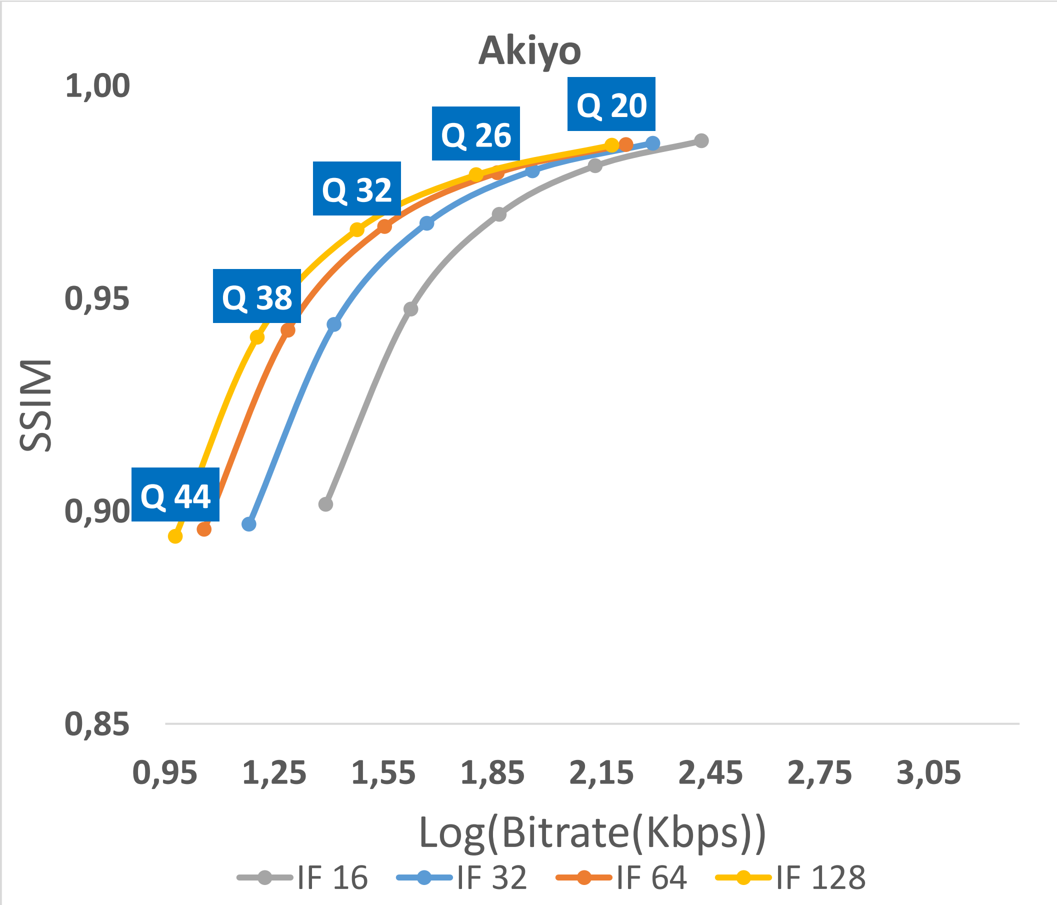

Figure 4.41. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.1.2. HEVC Bus

Table 4.58. *HEVC Bus PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	39.78	35.59	31.54	27.99	25.00
I Frame 32	39.58	35.26	31.14	27.56	24.59
I Frame 64	39.51	35.10	30.89	27.27	24.29
I Frame 128	39.46	34.98	30.68	27.03	24.02

Table 4.59. *HEVC Bus SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.98	0.97	0.93	0.85	0.72
I Frame 32	0.98	0.96	0.92	0.83	0.71
I Frame 64	0.98	0.96	0.91	0.82	0.69
I Frame 128	0.98	0.96	0.90	0.80	0.68

Table 4.60. *HEVC Bus bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	1765.10	827.86	378.10	173.47	79.92
I Frame 32	1670.09	749.00	323.13	140.01	62.90
I Frame 64	1631.86	715.81	300.00	126.11	55.43
I Frame 128	1617.14	701.65	288.33	118.00	51.86

Bus video has the highest TI video amongst all CIF videos. This complexity affects the output of I frame interval change as well. The gap between videos get smaller especially for lower QP values, it can be seen from Figure 4.42. However, for higher QP values I frame 16 configuration results way better quality value as well as higher bitrate values than other configurations. Table 4.58, Table 4.59 and Table 4.60 give the numerical values and I frame 128 has the lowest bitrate as well as quality values for all QP values.

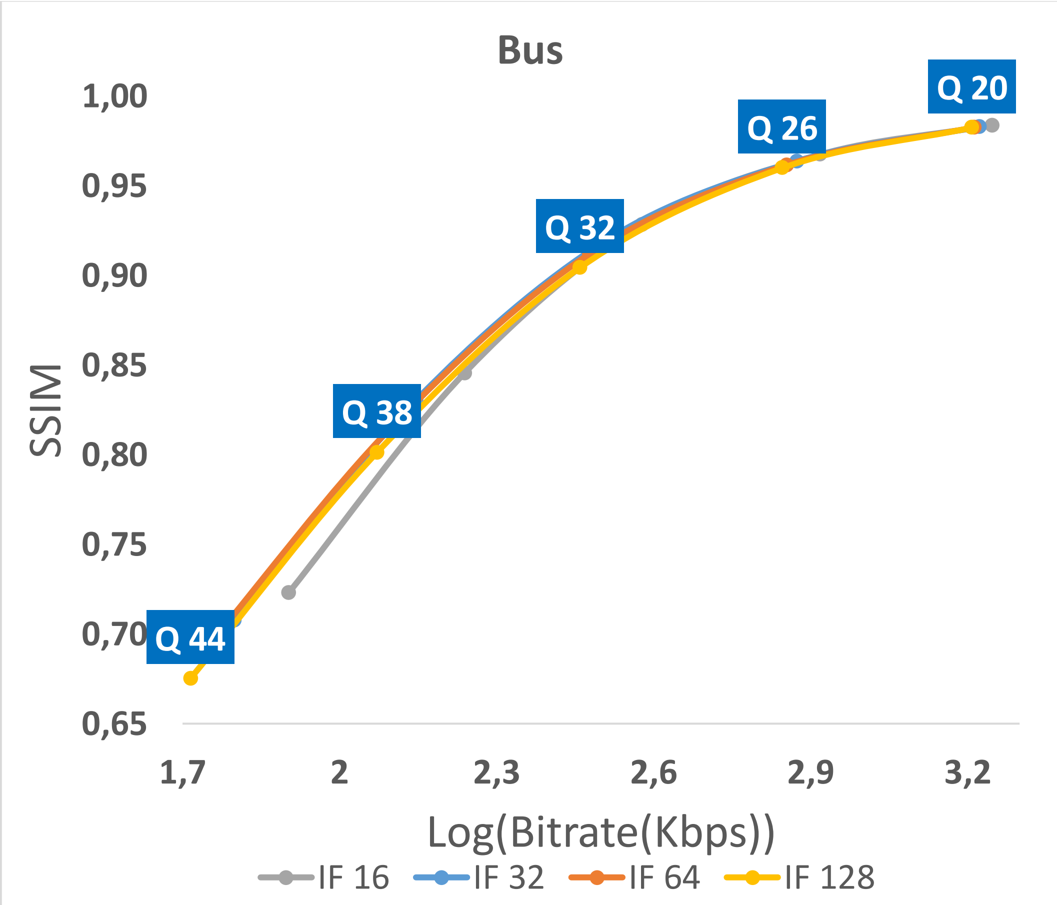

Figure 4.42. Bus SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.1.3. HEVC Container

Table 4.61. *HEVC Container PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	42.51	39.40	36.13	32.76	29.25
I Frame 32	42.11	38.80	35.51	32.26	28.85
I Frame 64	41.94	38.52	35.15	31.92	28.55
I Frame 128	41.83	38.32	34.87	31.65	28.28

Table 4.62. *HEVC Container SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.97	0.95	0.92	0.88	0.83
I Frame 32	0.97	0.95	0.91	0.87	0.83
I Frame 64	0.97	0.94	0.91	0.87	0.82
I Frame 128	0.97	0.94	0.90	0.86	0.82

Table 4.63. *HEVC Container bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	553.40	279.66	145.25	76.72	40.03
I Frame 32	421.39	185.26	86.11	43.39	22.79
I Frame 64	369.09	147.78	62.90	30.16	15.95
I Frame 128	342.56	129.16	51.44	23.78	12.62

Container video shows similar results as it is evident from Figure 4.43. I frame interval 128 configuration has lowest bitrate as well worst quality outputs, while I frame interval 16 configuration results best quality output as well as highest bitrate.

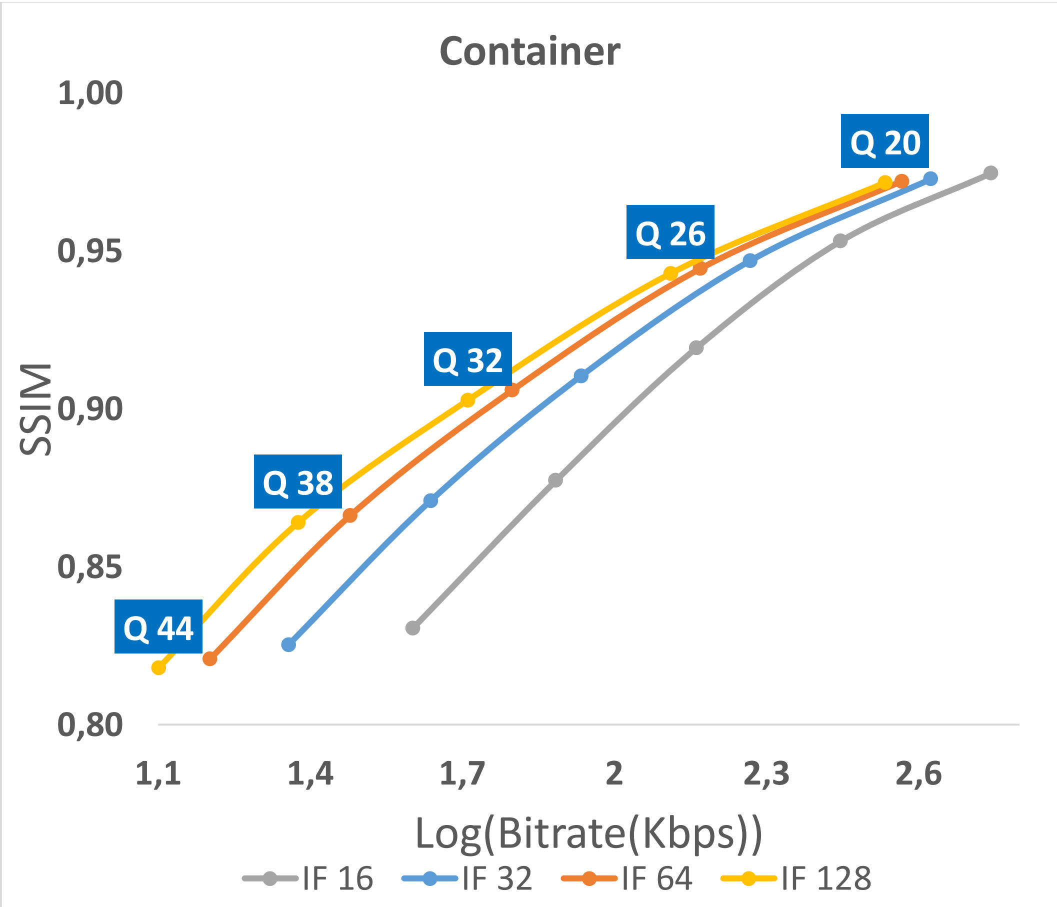

Figure 4.43. Container SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.1.4. HEVC Foreman

Table 4.64. *HEVC Foreman PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	41.76	38.53	35.07	31.81	28.86
I Frame 32	41.50	38.20	34.81	31.57	28.59
I Frame 64	41.39	38.02	34.62	31.38	28.39
I Frame 128	41.32	37.89	34.48	31.21	28.23

Table 4.65. *HEVC Foreman SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.98	0.96	0.93	0.88	0.82
I Frame 32	0.98	0.96	0.92	0.87	0.81
I Frame 64	0.98	0.96	0.92	0.87	0.80
I Frame 128	0.98	0.95	0.92	0.87	0.80

Table 4.66. *HEVC Foreman bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	776.12	341.48	150.47	70.32	35.80
I Frame 32	668.18	275.60	119.34	54.78	27.23
I Frame 64	624.60	251.04	106.69	48.71	24.16
I Frame 128	601.12	237.61	99.44	44.73	22.05

For video Foreman the situation is same. Table 4.64, Table 4.65 and Table 4.66 indicates I frame interval 128 configuration again results worst quality as well as lowest bitrate values and I frame interval 16 configuration results best quality as well as highest bitrate values.

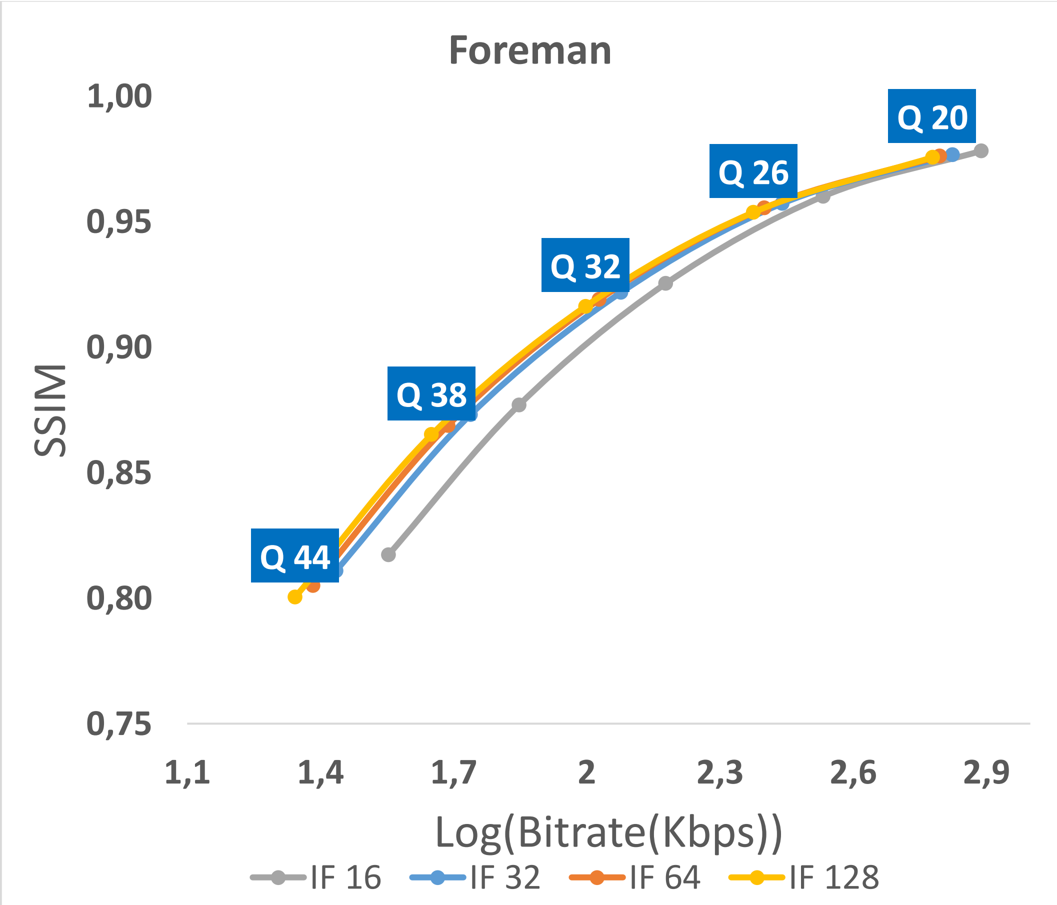

Figure 4.44. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.1.5. HEVC Mobile

Table 4.67. *HEVC Mobile PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	39.33	34.49	30.75	27.70	24.22
I Frame 32	39.15	34.22	30.33	27.16	23.77
I Frame 64	39.08	34.10	30.12	26.83	23.45
I Frame 128	39.04	34.03	30.01	26.61	23.20

Table 4.68. *HEVC Mobile SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.98	0.96	0.92	0.82
I Frame 32	0.99	0.98	0.95	0.91	0.81
I Frame 64	0.99	0.98	0.95	0.90	0.79
I Frame 128	0.99	0.98	0.95	0.90	0.78

Table 4.69. *HEVC Mobile bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	2408.33	1051.60	488.38	264.00	136.27
I Frame 32	2206.23	879.71	356.90	173.35	86.94
I Frame 64	2129.44	814.00	306.99	139.20	67.39
I Frame 128	2091.71	782.33	282.74	122.30	57.41

For video Mobile, the results are like other CIF quality videos. Figure 4.45 demonstrates the difference between configurations. Same as other videos I frame interval 128 configuration has the worst quality result as well as lowest bitrate value. When QP value increases the gap between configurations again increases.

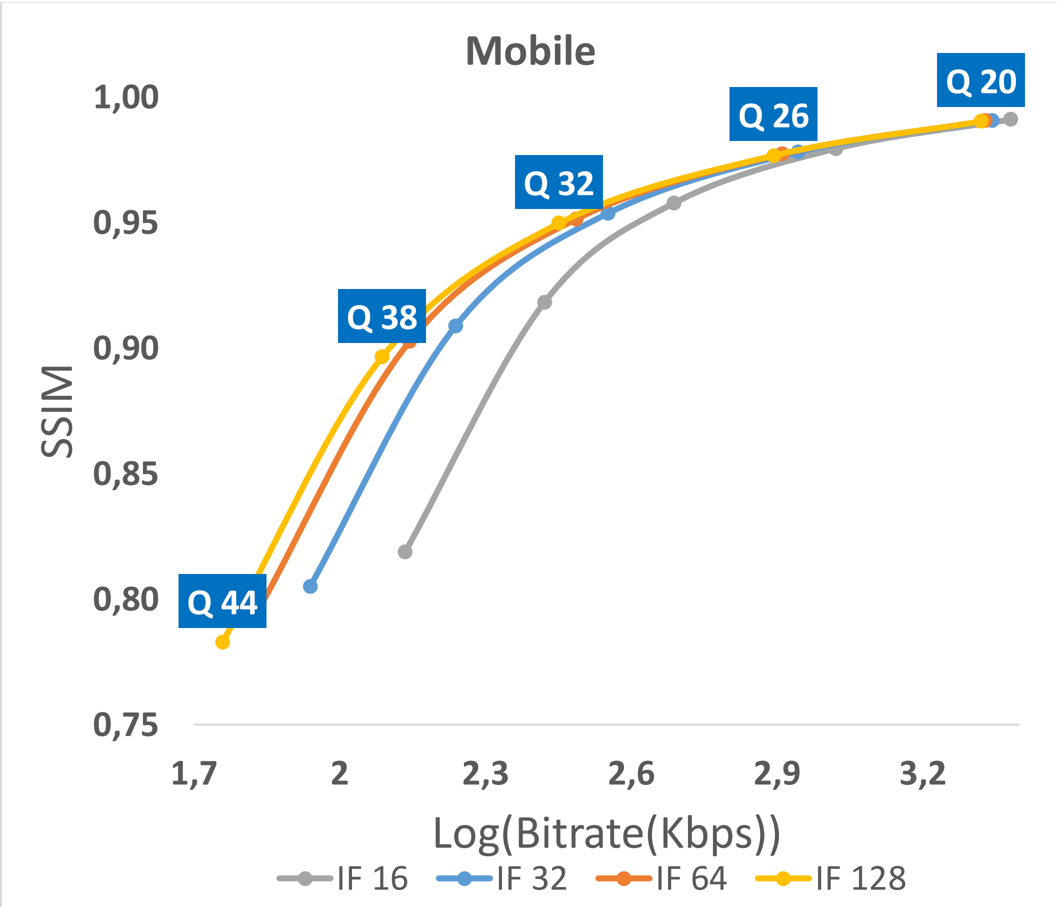

Figure 4.45. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.1.6. HEVC Basketball

Table 4.70. *HEVC Basketball PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	47.79	46.18	43.58	40.43	37.33
I Frame 32	47.67	45.99	43.35	40.17	37.03
I Frame 64	47.63	45.90	43.24	40.05	36.90
I Frame 128	47.62	45.88	43.22	40.03	36.89

Table 4.71. *HEVC Basketball SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.99	0.98	0.98	0.97
I Frame 32	0.99	0.99	0.98	0.98	0.96
I Frame 64	0.99	0.99	0.98	0.98	0.96
I Frame 128	0.99	0.99	0.98	0.98	0.96

Table 4.72. *HEVC Basketball bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	12056.68	4330.62	2052.31	1078.62	608.66
I Frame 32	10145.35	3609.54	1723.76	906.47	514.07
I Frame 64	9257.87	3292.37	1578.91	830.32	472.03
I Frame 128	8781.69	3138.50	1507.50	792.57	450.87

4K video Basketball shows the similar results as in CIF video cases. Since the dimensions of a video does not have an influence of inter or intra coding principle of frames, this is a very expected result. As in all CIF videos I frame interval 128 configuration has the lowest bitrate value as well as worst PSNR output, while decreasing I frame interval increases the PSNR value as well as increase the bitrate.

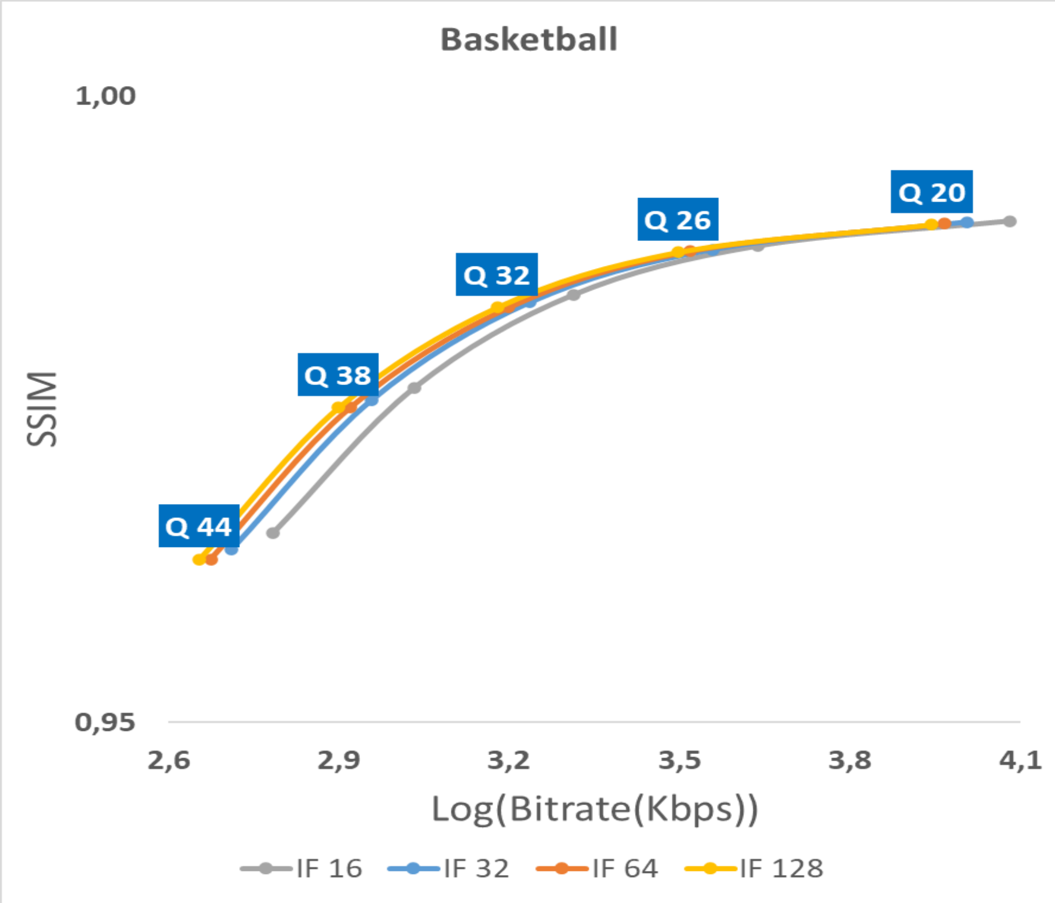

Figure 4.46. Basketball SSIM vs Log(Bitrate(Kbps) graph for different CTU and depth sizes

4.4.1.7. HEVC BD Rate results

HEVC I Frame difference results show that for all cases using smallest I frame difference gives the best result in terms of quality; however, it also costs the most bitrate among all of them. When the BD rates are examined from Figure 4.47, the best overall results are acquired for the case when I frame interval is largest which is 128. The quality may be worse than other I frame configurations; however, it saves way too much bitrate. Especially when video complexity gets lower, the performance of I frame interval 128 configuration gets even better than other configurations.

Figure 4.47 shows that for lower temporal information videos such as Akiyo and Container, efficiency of selecting I frame interval 128 configuration is higher than other videos. Selecting I frame interval 128 configuration results more than %50 percent bitrate save at the same quality level than I frame interval 16 configuration for videos Akiyo and Container. This happens because when encoding higher complexity videos, creating I frame, which has the full information of the frame, become more important to efficiently decode other B frames, selecting higher I frame intervals for video Bus results less efficiency than lower temporal behavior videos since Bus needs more frequent I frames to encode B frames more efficiently. However, for video Akiyo all the frames have almost same data, only small part of the video is changing, therefore having an I frame less frequently improves the bitrate efficiency more than other videos.

The results show similarity for 4K video as well. I frame interval 128 configuration has lowest bitrate results as well as worse quality outputs than other I frame configurations. However, I frame interval 128 configuration has more efficiency than other configurations. It saves more bitrate than others. As BD rates from Figure 4.47 shows, I frame interval 128 configuration saves almost %20 bitrate at the same quality level than I frame interval 16 configuration. Since video Basketball have high temporal behavior, the efficiency is not as high as in videos Akiyo and Container.

HEVC I frame difference tests result expected outputs. The best quality levels are acquired from I frame interval 16 configuration as well as it results highest bitrate values. However, the best performance is obtained by choosing the I frame difference as maximum as it can be, while the quality decreases when increasing the I frame interval, the reduction in bitrate has more value. Therefore, choosing largest possible I frame interval while selecting 1 or 2 less QP levels gives way better results at similar bitrate levels.

			IF 16	IF 32	IF 64	IF 128
Akiyo	IF 16	BDBR		-30,82	-44,77	-51,33
		BDPSNR		1,91	2,94	3,44
	IF 32	BDBR	44,55		-20,18	-29,67
		BDPSNR	-1,91		1,11	1,67
	IF 64	BDBR	81,07	25,28		-11,87
		BDPSNR	-2,94	-1,11		0,60
	IF 128	BDBR	105,47	42,19	13,47	
		BDPSNR	-3,44	-1,67	-0,60	
			IF 16	IF 32	IF 64	IF 128
Bus	IF 16	BDBR		-6,53	-8,39	-8,13
		BDPSNR		0,32	0,41	0,40
	IF 32	BDBR	6,99		-2,01	-1,72
		BDPSNR	-0,32		0,09	0,08
	IF 64	BDBR	9,15	2,05		0,32
		BDPSNR	-0,41	-0,09		-0,01
	IF 128	BDBR	8,85	1,74	-0,31	
		BDPSNR	-0,40	-0,08	0,01	
			IF 16	IF 32	IF 64	IF 128
Container	IF 16	BDBR		-31,33	-44,87	-51,41
		BDPSNR		1,74	2,58	2,99
	IF 32	BDBR	45,62		-19,74	-29,29
		BDPSNR	-1,74		0,93	1,40
	IF 64	BDBR	81,40	24,59		-11,94
		BDPSNR	-2,58	-0,93		0,51
	IF 128	BDBR	105,81	41,42	13,56	
		BDPSNR	-2,99	-1,40	-0,51	
			IF 16	IF 32	IF 64	IF 128
Foreman	IF 16	BDBR		-14,85	-19,77	-22,38
		BDPSNR		0,66	0,90	1,02
	IF 32	BDBR	17,43		-5,74	-8,82
		BDPSNR	-0,66		0,24	0,37
	IF 64	BDBR	24,64	6,09		-3,28
		BDPSNR	-0,90	-0,24		0,13
	IF 128	BDBR	28,84	9,67	3,39	
		BDPSNR	-1,02	-0,37	-0,13	
			IF 16	IF 32	IF 64	IF 128
Mobile	IF 16	BDBR		-18,03	-25,00	-28,38
		BDPSNR		0,94	1,29	1,46
	IF 32	BDBR	21,99		-8,71	-12,95
		BDPSNR	-0,94		0,41	0,60
	IF 64	BDBR	33,33	9,54		-4,74
		BDPSNR	-1,29	-0,41		0,21
	IF 128	BDBR	39,63	14,87	4,98	
		BDPSNR	-1,46	-0,60	-0,21	
			IF 16	IF 32	IF 64	IF 128
Basketball	IF 16	BDBR		-10,98	-16,34	-19,87
		BDPSNR		0,42	0,64	0,80
	IF 32	BDBR	12,34		-5,98	-9,95
		BDPSNR	-0,42		0,22	0,38
	IF 64	BDBR	19,53	6,36		-4,23
		BDPSNR	-0,64	-0,22		0,15
	IF 128	BDBR	24,80	11,05	4,41	
		BDPSNR	-0,80	-0,38	-0,15	

Figure 4.47. HEVC BD Rates of all videos for different CTU and depth sizes

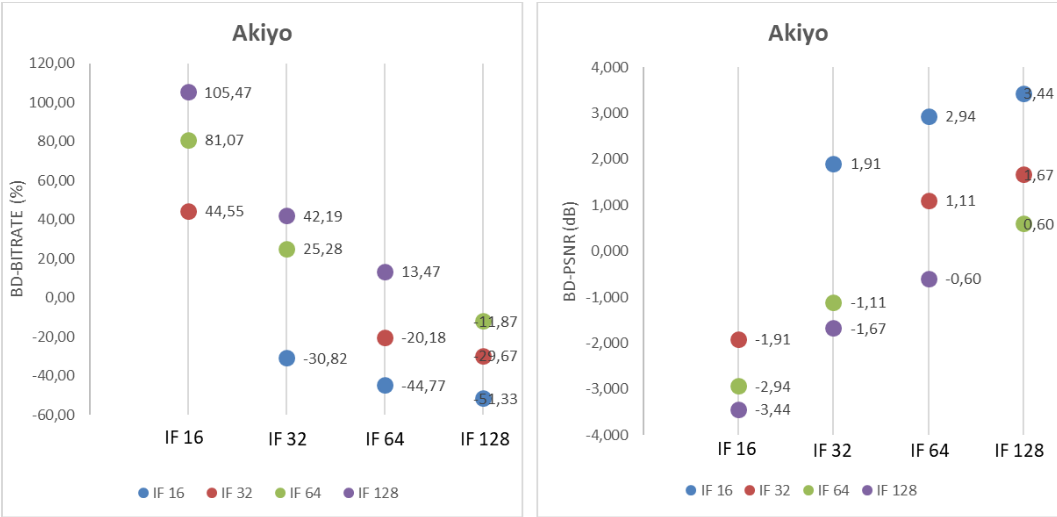

Figure 4.48. Akiyo BD Rates for HEVC

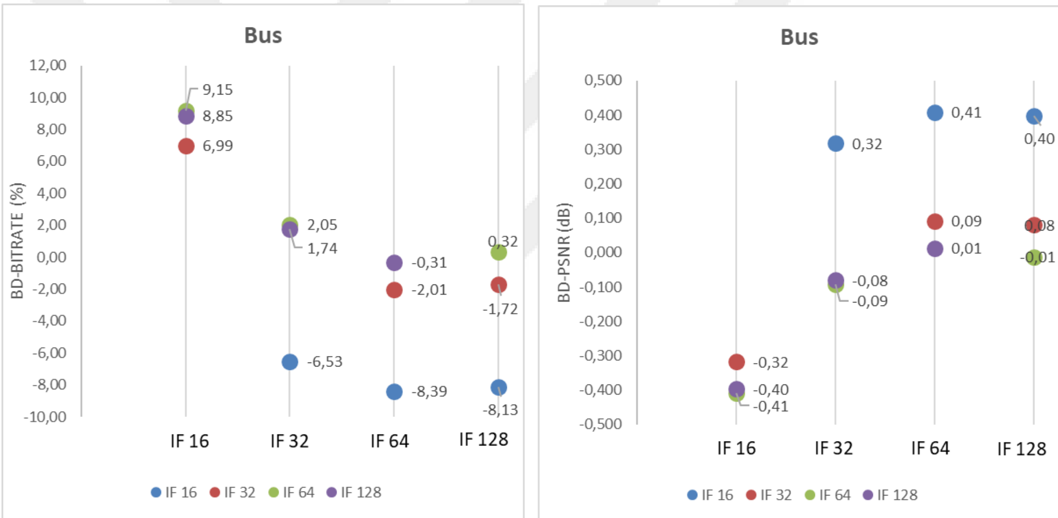

Figure 4.49. Bus BD Rates for HEVC

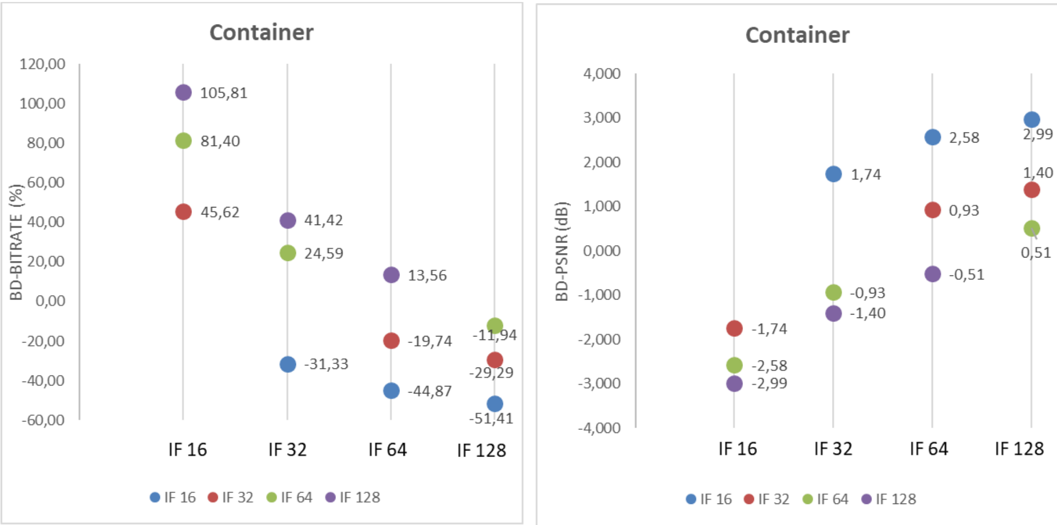

Figure 4.50. Container BD Rates for HEVC

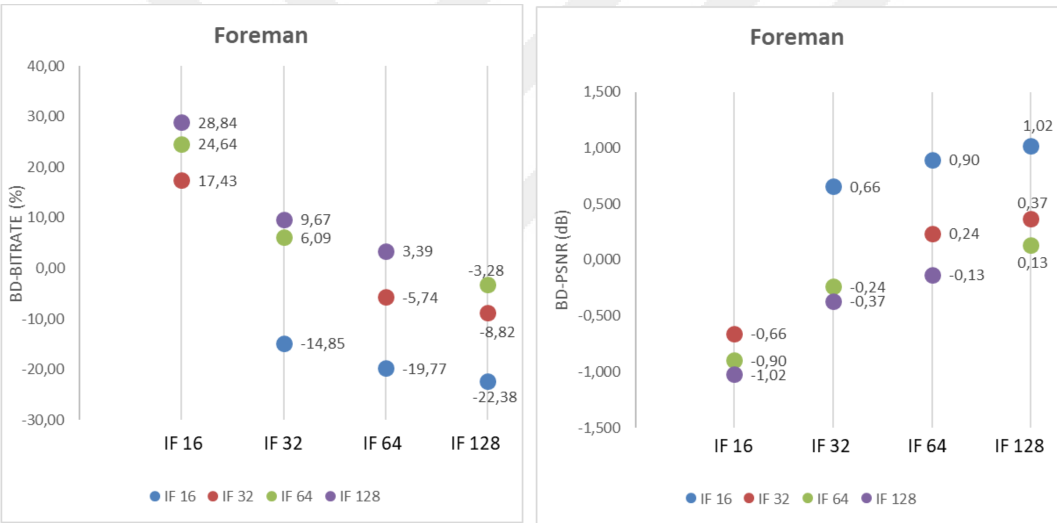

Figure 4.51. Foreman BD Rates for HEVC

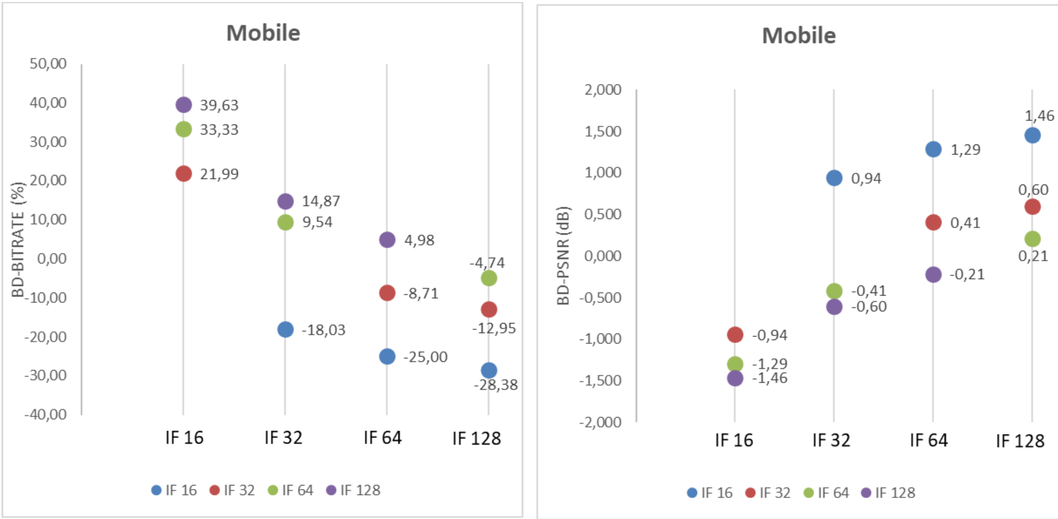

Figure 4.52. Mobile BD Rates for HEVC

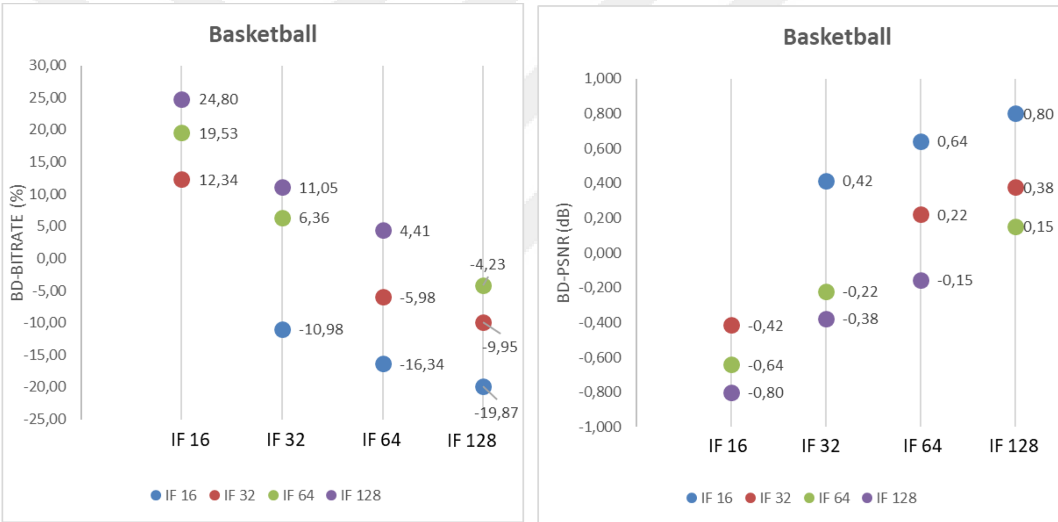

Figure 4.53. Basketball BD Rates for HEVC

4.4.2. VVC I Frame Difference

In the configuration file "IntraPeriod" parameters is adjusted at 16, 32, 64 and 128 while performing I frame difference test. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8. Quantization parameter is set to 20, 26, 32, 38, 44 while performing the tests. CTU size is adjusted at 128, and partition depth is fixed at 3. Rate control is off. Other parameters are all taken as default parameters. VVC codec uses all B frames for encoding process.

4.4.2.1. VVC Akiyo

Table 4.73. *VVC Akiyo PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	45.16	43.12	40.42	37.28	33.89
I Frame 32	44.95	42.77	39.93	36.82	33.43
I Frame 64	44.85	42.58	39.71	36.58	33.19
I Frame 128	44.74	42.43	39.48	36.40	32.94

Table 4.74. *VVC Akiyo SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.98	0.97	0.96	0.93
I Frame 32	0.99	0.98	0.97	0.96	0.92
I Frame 64	0.99	0.98	0.97	0.95	0.92
I Frame 128	0.99	0.98	0.97	0.95	0.92

Table 4.75 *VVC Akiyo bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	227.33	122.55	67.44	37.28	24.76
I Frame 32	167.02	83.91	42.71	24.37	15.64
I Frame 64	141.72	67.80	32.87	18.37	11.87
I Frame 128	128.88	60.02	27.67	15.41	10.03

VVC adopts same principle for I frame interval and GOP size difference as in HEVC. VVC brings up some new intra and inter coding techniques; however, the improvement does not change the behavior of the VVC codec for I frame interval change. Figure 4.54 shows the results for video Akiyo and it has similar type of RD curves when compared with HEVC. Best quality acquired when I frame interval is minimum which is 16 as well as highest bitrate value is obtained from I frame interval 16 configuration for all QP values. I frame interval 128 configuration has the worst quality value as well as lowest bitrate value.

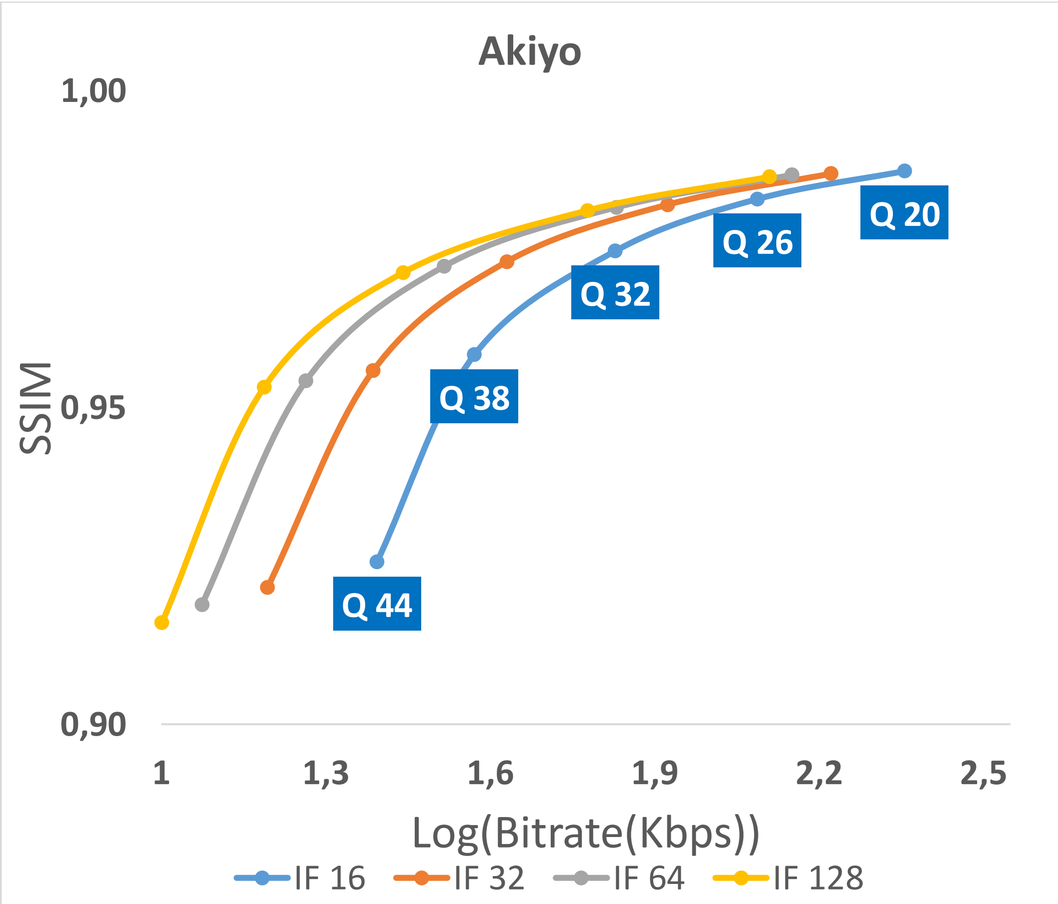

Figure 4.54. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.2.2. VVC Bus

Table 4.76. *VVC Bus PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	40.36	36.30	32.35	28.75	25.59
I Frame 32	40.16	35.97	31.96	28.37	25.27
I Frame 64	40.07	35.78	31.68	28.04	24.93
I Frame 128	40.04	35.65	31.45	27.76	24.66

Table 4.77. *VVC Bus SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.97	0.94	0.87	0.75
I Frame 32	0.98	0.97	0.93	0.85	0.73
I Frame 64	0.98	0.97	0.93	0.84	0.72
I Frame 128	0.98	0.97	0.92	0.82	0.70

Table 4.78 *VVC Bus bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	1488.17	689.12	319.04	146.89	66.12
I Frame 32	1393.04	609.24	262.59	113.70	49.45
I Frame 64	1355.25	577.21	239.40	99.23	42.15
I Frame 128	1339.04	562.94	226.81	92.04	38.21

In Bus video as well, the behavior is very similar between HEVC and VVC. I frame interval 128 configuration has the lowest bitrate value as well as lowest quality output for all QP values again.

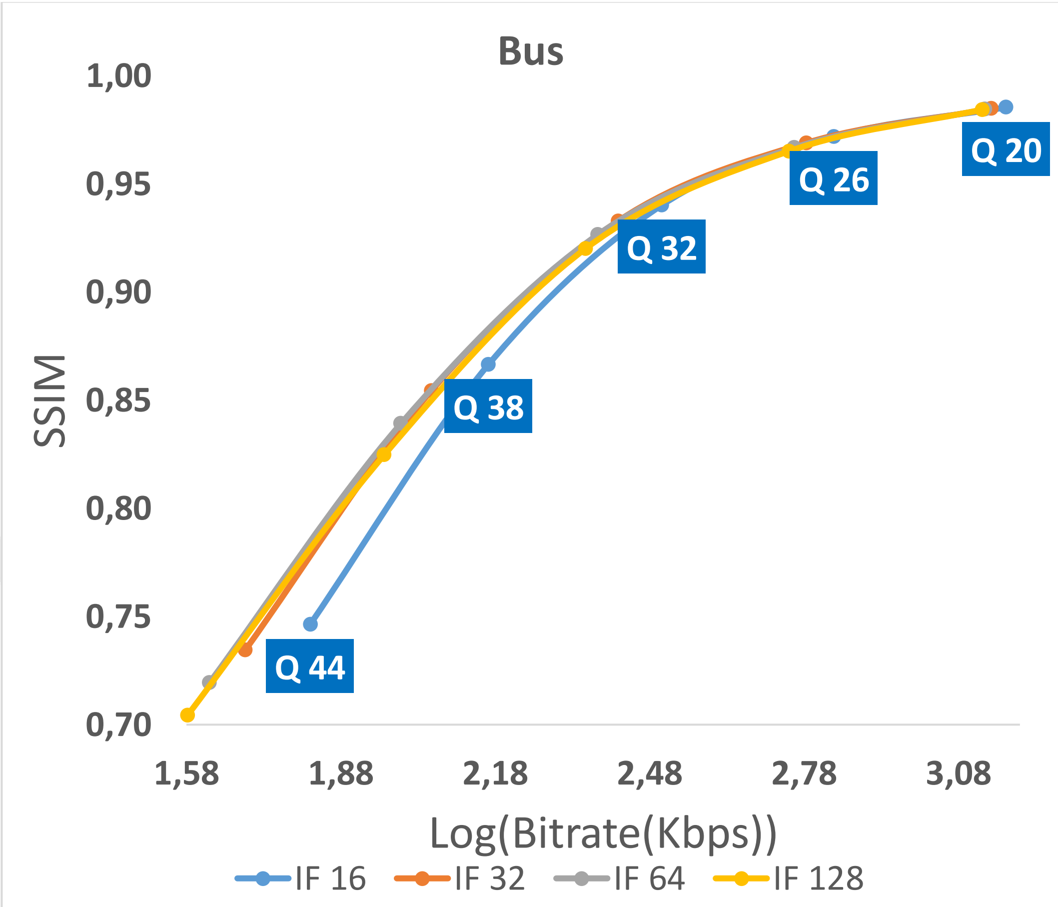

Figure 4.55. Bus SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.2.3. VVC Container

Table 4.79. *VVC Container PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	42.93	39.91	36.72	33.47	29.99
I Frame 32	42.53	39.30	36.09	32.96	29.64
I Frame 64	42.37	38.98	35.74	32.61	29.32
I Frame 128	42.26	38.76	35.43	32.26	28.98

Table 4.80. *VVC Container SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.98	0.96	0.93	0.89	0.84
I Frame 32	0.98	0.95	0.92	0.88	0.84
I Frame 64	0.97	0.95	0.91	0.88	0.83
I Frame 128	0.97	0.95	0.91	0.87	0.83

Table 4.81 *VVC Container bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	513.07	256.86	133.46	70.15	36.67
I Frame 32	391.55	169.71	78.59	39.76	21.16
I Frame 64	344.25	134.49	56.63	27.60	14.82
I Frame 128	320.15	117.15	45.86	21.71	11.74

Container video shows similar results as others, best results are obtained when I frame interval is selected as minimum.

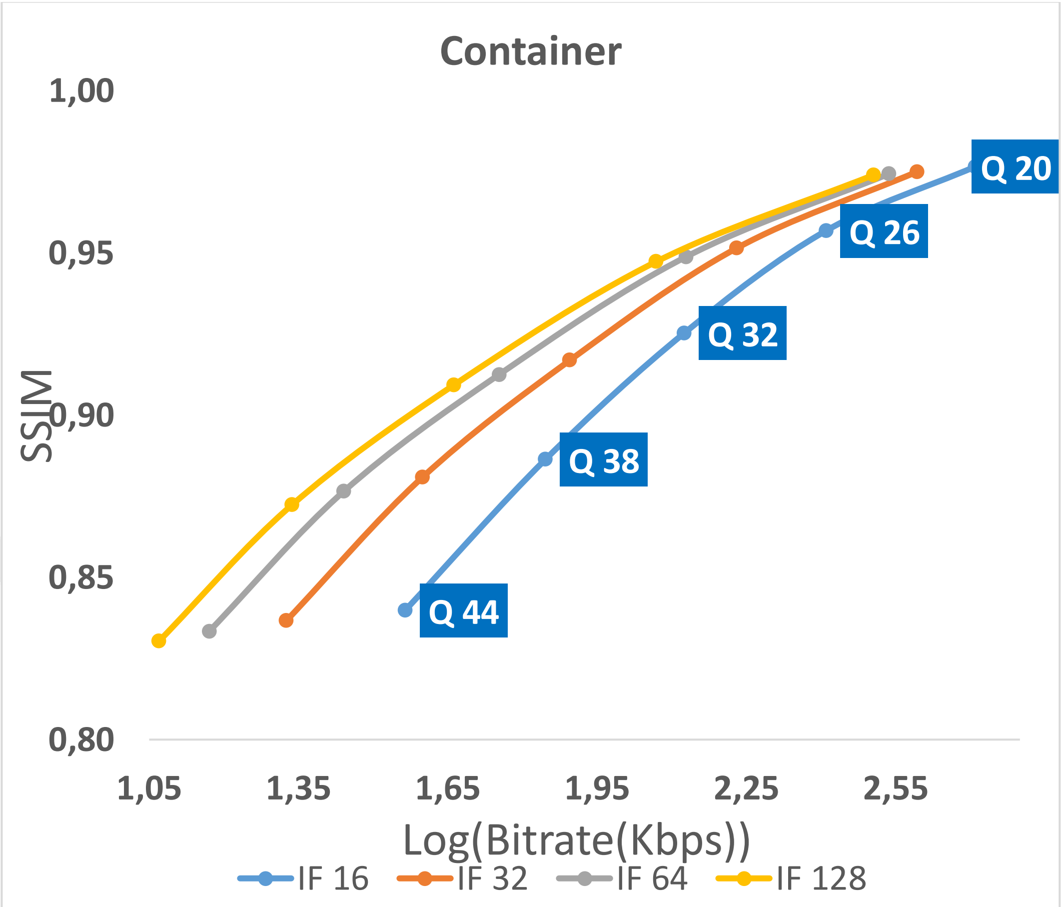

Figure 4.56. Container SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.2.4. VVC Foreman

Table 4.82. *VVC Foreman PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	42.00	39.08	35.80	32.62	29.65
I Frame 32	41.69	38.76	35.52	32.37	29.43
I Frame 64	41.56	38.53	35.29	32.16	29.20
I Frame 128	41.47	38.37	35.10	32.00	29.04

Table 4.83. *VVC Foreman SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.98	0.97	0.94	0.89	0.84
I Frame 32	0.98	0.96	0.93	0.89	0.83
I Frame 64	0.98	0.96	0.93	0.88	0.83
I Frame 128	0.98	0.96	0.93	0.88	0.82

Table 4.84 *VVC Foreman bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	619.86	280.82	126.61	60.08	31.45
I Frame 32	521.24	223.26	97.33	45.57	23.60
I Frame 64	480.52	198.99	85.35	39.77	20.51
I Frame 128	458.11	186.74	78.94	36.38	18.72

Foreman video in VVC shows similar results as it can be seen from Figure 4.57.

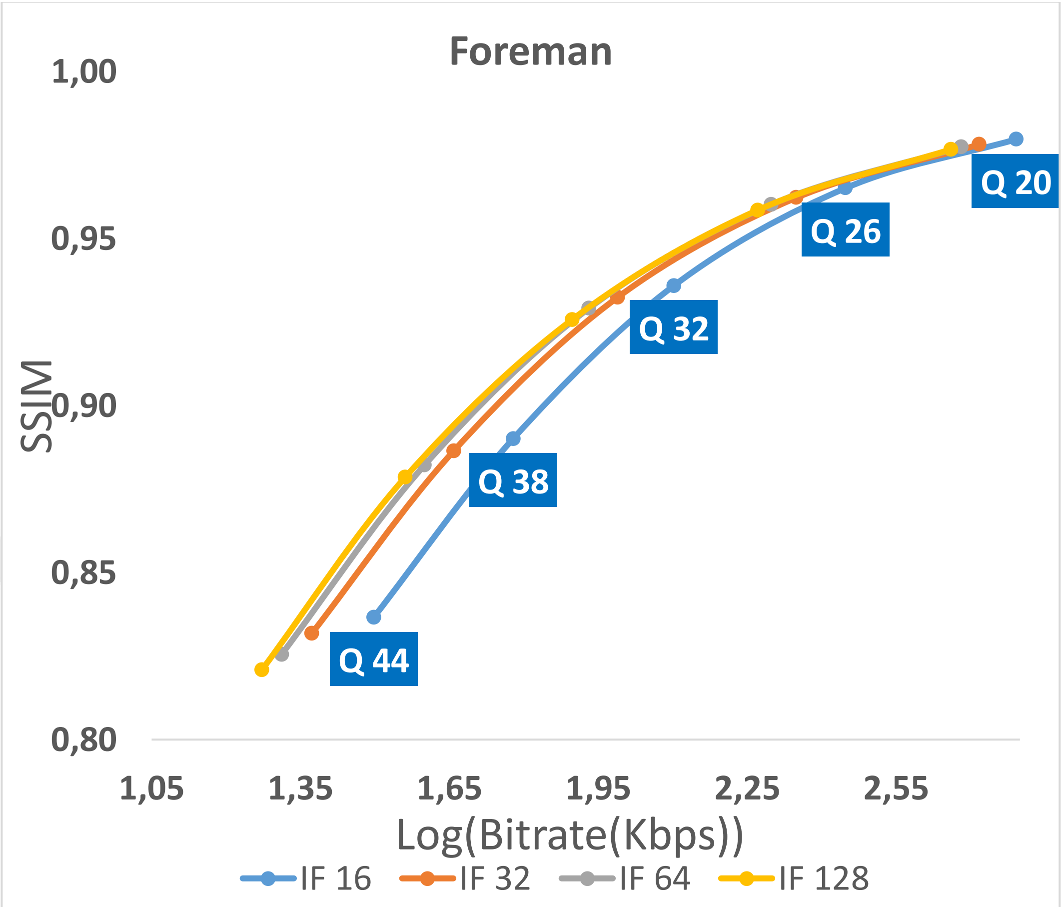

Figure 4.57. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.2.5. VVC Mobile

Table 4.85. *VVC Mobile PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	39.66	34.88	31.29	28.47	25.13
I Frame 32	39.45	34.57	30.82	27.92	24.65
I Frame 64	39.36	34.43	30.60	27.57	24.32
I Frame 128	39.31	34.36	30.46	27.31	24.09

Table 4.86. *VVC Mobile SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.98	0.96	0.93	0.86
I Frame 32	0.99	0.98	0.96	0.93	0.84
I Frame 64	0.99	0.98	0.96	0.92	0.83
I Frame 128	0.99	0.98	0.96	0.91	0.82

Table 4.87 *VVC Mobile bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	1998.68	852.14	410.93	221.68	115.11
I Frame 32	1805.98	693.52	282.50	136.12	68.85
I Frame 64	1773.76	630.77	231.78	102.44	50.43
I Frame 128	1697.85	599.02	206.16	85.68	40.99

For Mobile video, there is no difference from other videos. I frame interval 128 configuration has lowest bitrate values as well as worst quality values for all QP levels.

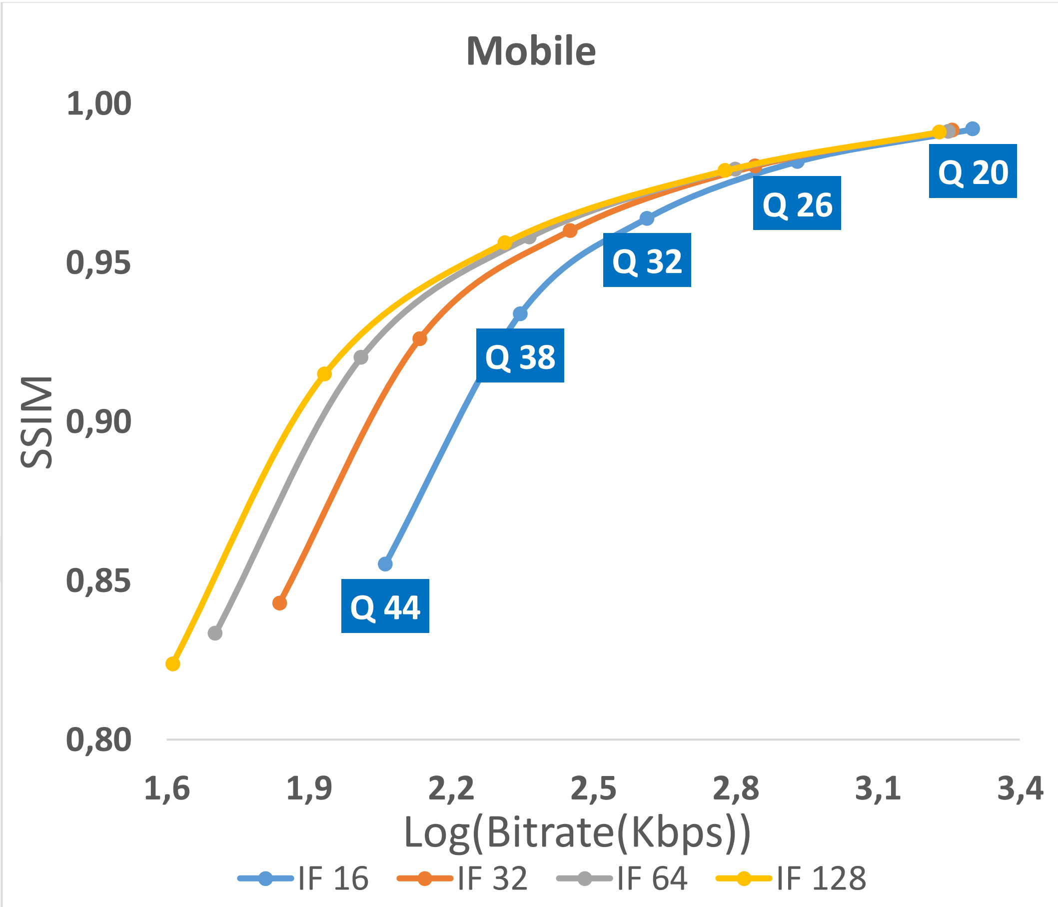

Figure 4.58. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.2.6. VVC Basketball

Table 4.88. *VVC Basketball PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	48.68	47.21	44.71	41.66	38.22
I Frame 32	48.59	47.04	44.51	41.44	37.97
I Frame 64	48.56	46.95	44.42	41.33	37.87
I Frame 128	48.56	46.94	44.42	41.31	37.86

Table 4.89. *VVC Basketball SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	0.99	0.99	0.99	0.98	0.97
I Frame 32	0.99	0.99	0.99	0.98	0.97
I Frame 64	0.99	0.99	0.99	0.98	0.97
I Frame 128	0.99	0.99	0.99	0.98	0.97

Table 4.90 *VVC Basketball bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
I Frame 16	11451.53	3584.49	1658.78	867.13	455.48
I Frame 32	9345.16	2918.40	1362.22	710.63	370.73
I Frame 64	8429.33	2626.51	1233.83	643.09	335.03
I Frame 128	7947.98	2489.81	1172.30	610.06	317.26

For video Basketball the results are very similar to HEVC. Figure 4.59 shows that using lower I frame interval results better quality values as well as higher bitrates for all QP values.

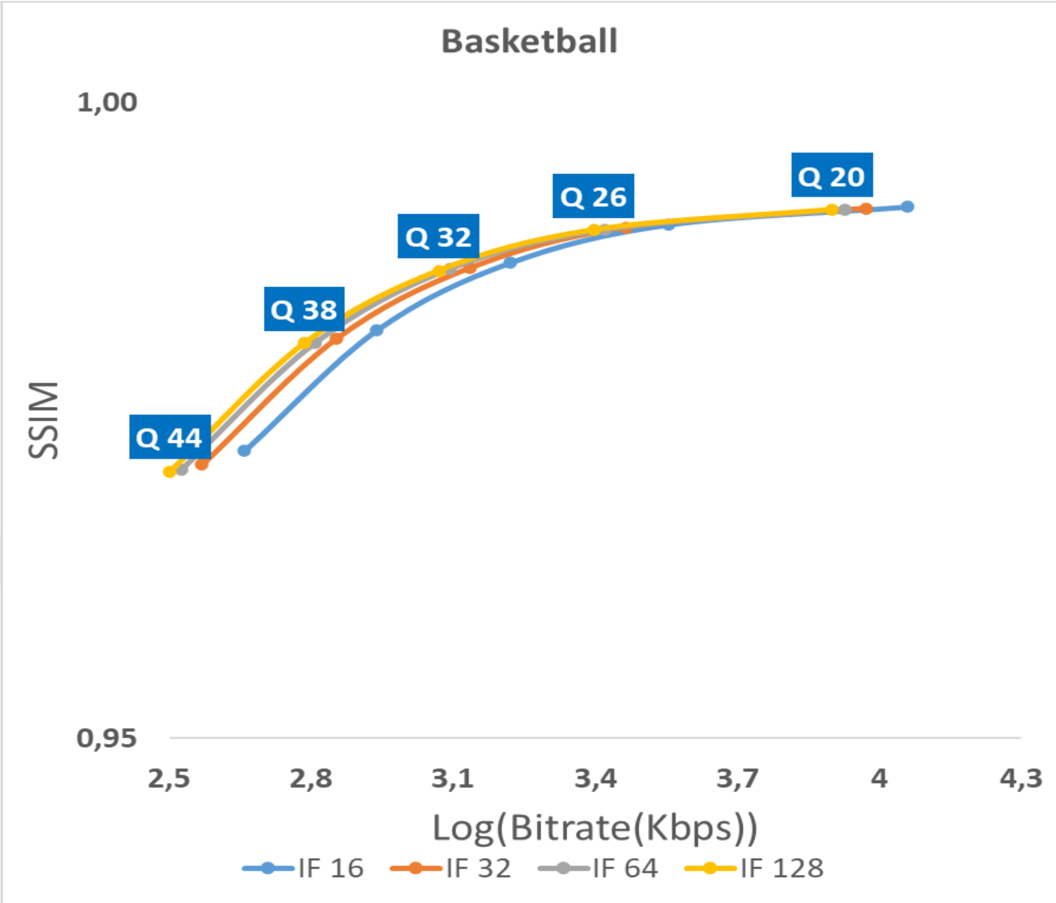

Figure 4.59. Basketball SSIM vs Log(Bitrate(Kbps) graph for different CTU and depth sizes

4.4.2.7. VVC BD Rate results

VVC shows exactly same behavior with HEVC I frame configurations. I frame interval 128 configuration results worst quality levels as well as lowest bitrate values, while I frame interval 16 configuration results best quality levels and highest bitrate values. However, the efficiency of using larger I frame interval configuration sizes is higher than using lower I frame interval configuration.

In CIF quality videos for lower complexity videos such as Akiyo and Container the efficiency of using large I frame interval is higher than more complex videos. As in HEVC using less I frame for low temporal behavior videos is more efficient, since introducing I frame more frequently brings large bitrate increase, while increasing quality very little. For Akiyo and Container, I frame interval 128 configurations save more than %40 bitrate than I frame interval 16 configuration at the same quality level. For more complex videos such as Bus, this efficiency drops nearly to %13. Higher temporal behavior videos need more frequent I frame to encode more effectively.

For 4K video Basketball the results are same with CIF videos. Since it has decent temporal behavior the efficiency of using I frame interval 128 configuration is not as efficient as Akiyo or Container.

One important difference between VVC and HEVC is that in 4K video Basketball, VVC I frame interval 128 configuration performs percentagewise better than HEVC I frame interval 128. This is because VVC implements more sophisticated inter and intra predictions, it saves even more bitrate when compared with HEVC. Other than this the behavior of changing I frame interval for HEVC and VVC resembles to each other.

			IF 16	IF 32	IF 64	IF 128
Akiyo	IF 16	BDBR		-28,66	-42,11	-48,51
		BDPSNR		1,58	2,40	2,79
	IF 32	BDBR	40,17		-18,93	-27,91
		BDPSNR	-1,58		0,95	1,42
	IF 64	BDBR	72,76	23,35		-11,06
		BDPSNR	-2,40	-0,95		0,52
	IF 128	BDBR	94,20	38,71	12,44	
		BDPSNR	-2,79	-1,42	-0,52	
			IF 16	IF 32	IF 64	IF 128
Bus	IF 16	BDBR		-10,04	-12,71	-12,92
		BDPSNR		0,48	0,61	0,62
	IF 32	BDBR	11,16		-2,98	-3,23
		BDPSNR	-0,48		0,13	0,14
	IF 64	BDBR	14,56	3,08		-0,27
		BDPSNR	-0,61	-0,13		0,01
	IF 128	BDBR	14,84	3,34	0,27	
		BDPSNR	-0,62	-0,14	-0,01	
			IF 16	IF 32	IF 64	IF 128
Container	IF 16	BDBR		-31,07	-44,75	-51,01
		BDPSNR		1,67	2,48	2,85
	IF 32	BDBR	45,08		-19,84	-28,91
		BDPSNR	-1,67		0,91	1,34
	IF 64	BDBR	81,01	24,74		-11,36
		BDPSNR	-2,48	-0,91		0,47
	IF 128	BDBR	104,11	40,68	12,81	
		BDPSNR	-2,85	-1,34	-0,47	
			IF 16	IF 32	IF 64	IF 128
Foreman	IF 16	BDBR		-16,60	-22,35	-24,96
		BDPSNR		0,73	1,00	1,13
	IF 32	BDBR	19,91		-6,82	-9,94
		BDPSNR	-0,73		0,28	0,40
	IF 64	BDBR	28,77	7,32		-3,37
		BDPSNR	-1,00	-0,28		0,13
	IF 128	BDBR	33,27	11,04	3,49	
		BDPSNR	-1,13	-0,40	-0,13	
			IF 16	IF 32	IF 64	IF 128
Mobile	IF 16	BDBR		-20,63	-29,08	-33,78
		BDPSNR		1,04	1,43	1,66
	IF 32	BDBR	25,99		-10,94	-17,02
		BDPSNR	-1,04		0,48	0,75
	IF 64	BDBR	41,01	12,29		-6,97
		BDPSNR	-1,43	-0,48		0,29
	IF 128	BDBR	51,01	20,51	7,49	
		BDPSNR	-1,66	-0,75	-0,29	
			IF 16	IF 32	IF 64	IF 128
Basketball	IF 16	BDBR		-13,65	-20,04	-23,95
		BDPSNR		0,48	0,74	0,91
	IF 32	BDBR	15,80		-7,38	-11,91
		BDPSNR	-0,48		0,25	0,42
	IF 64	BDBR	25,07	7,97		-4,89
		BDPSNR	-0,74	-0,25		0,17
	IF 128	BDBR	31,49	13,52	5,14	
		BDPSNR	-0,91	-0,42	-0,17	

Figure 4.60. VVC BD Rates of all videos for different CTU and depth sizes

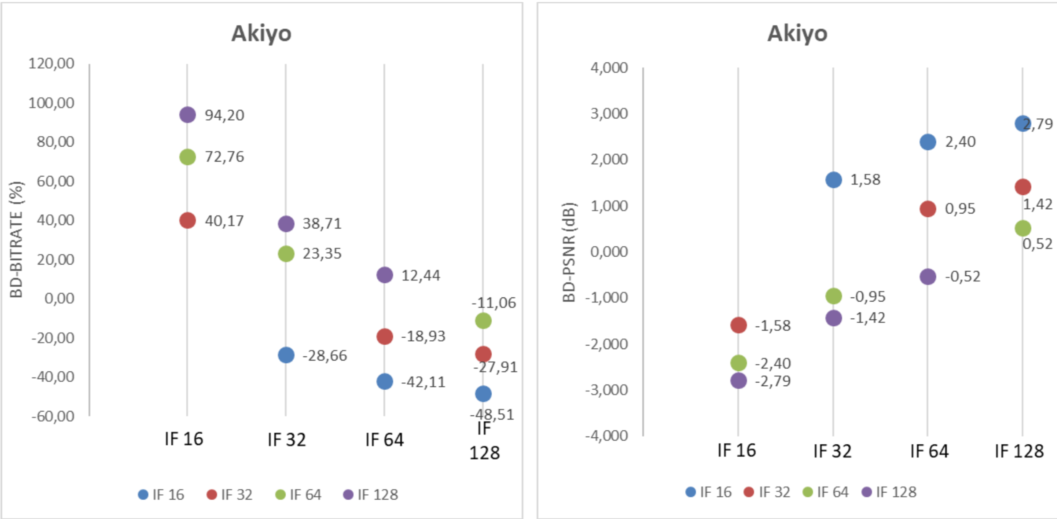

Figure 4.61. Akiyo BD Rates for VVC

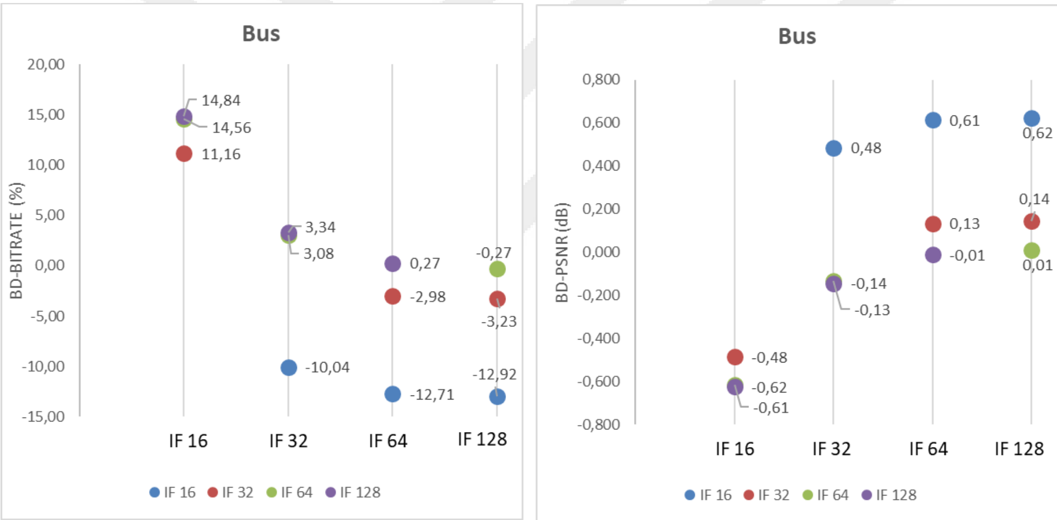

Figure 4.62. Bus BD Rates for VVC

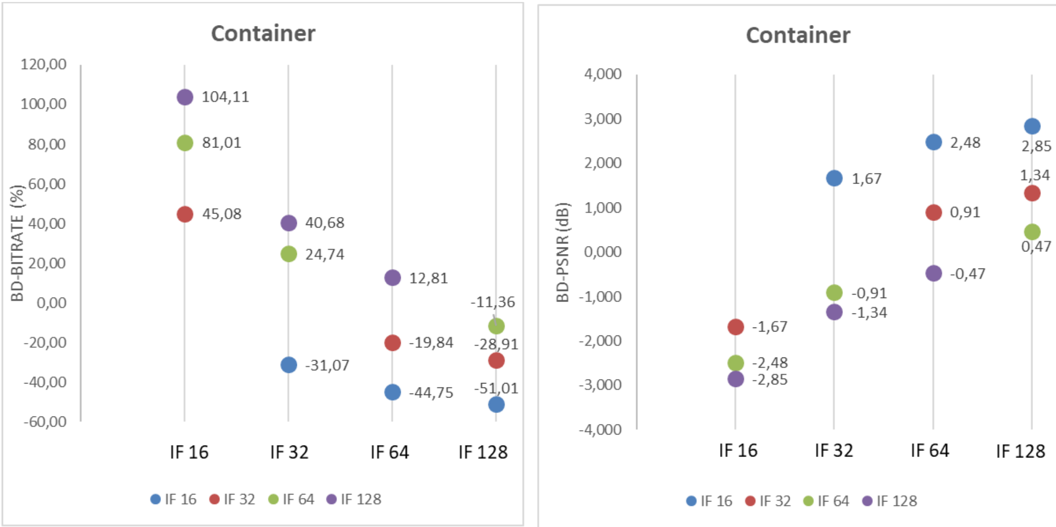

Figure 4.63. Container BD Rates for VVC

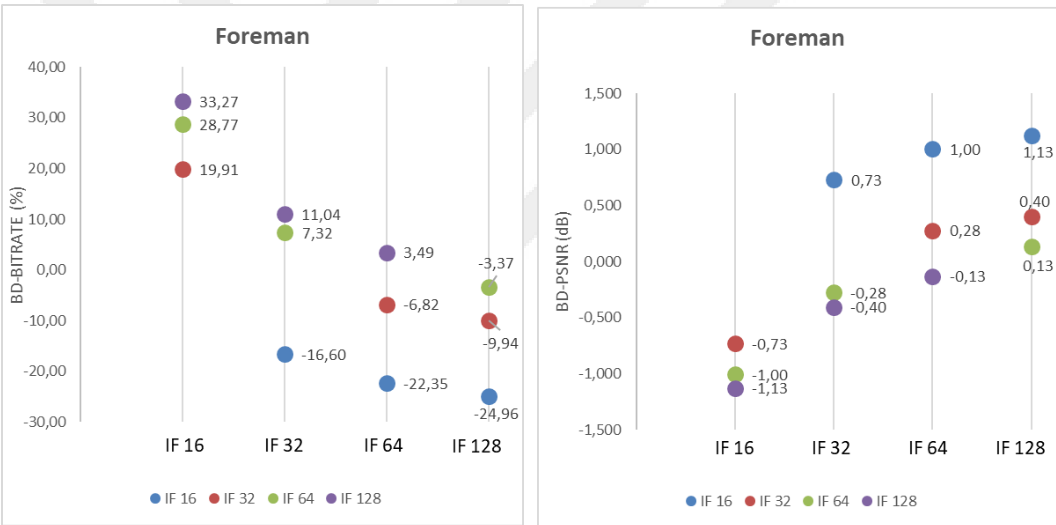

Figure 4.64. Foreman BD Rates for VVC

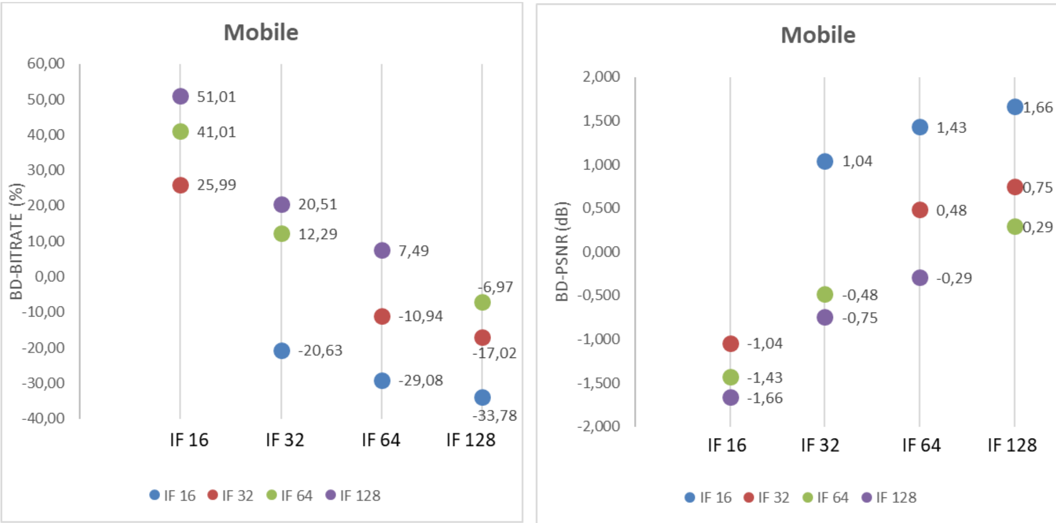

Figure 4.65. Mobile BD Rates for VVC

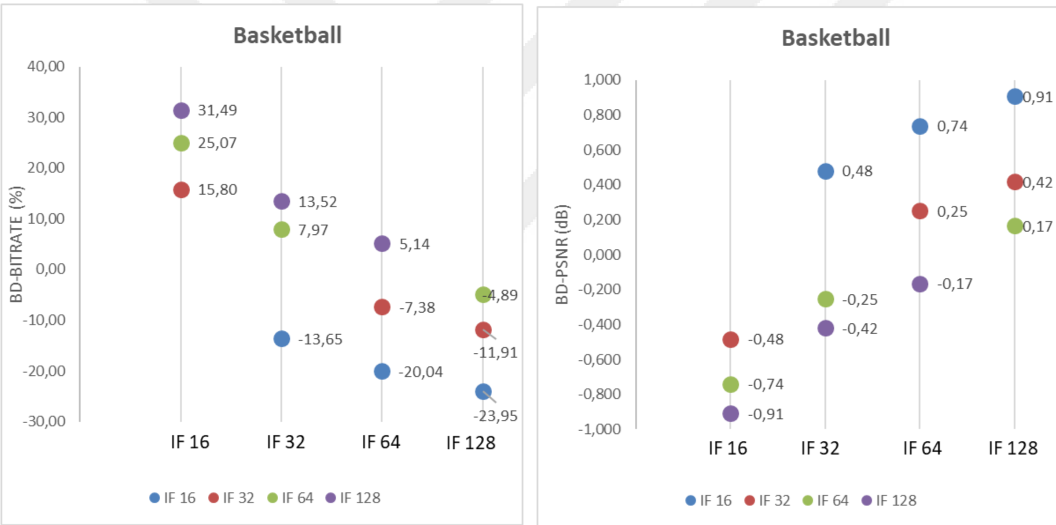

Figure 4.66. Basketball BD Rates for VVC

4.4.3. AV1 Golden Frame (GF) Interval

In the configuration file "--min-gf-interval" and "--max-gf-interval" parameters are set to 8, 16, 24. GF is the I frame of the AV1 codec, thus fixing its minimum and maximum values at the same value helps to compare AV1 with HEVC and VVC. Moreover, AV1 does not support high GF values such as 128. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8. Quantization parameter is set to 20, 26, 32, 38, 44 while performing the tests. "--min-partition-size" is set to 16 and "--max-partition-size" is set to 64.

4.4.3.1. AV1 Akiyo

Table 4.91. *AV1 Akiyo PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	46.10	44.57	42.97	41.42	39.92
GF Interval 16	47.97	46.79	45.46	44.10	42.81
GF Interval 8	48.40	47.12	45.75	44.58	43.37

Table 4.92. *AV1 Akiyo SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	0.99	0.99	0.98	0.98	0.97
GF Interval 16	0.99	0.99	0.99	0.99	0.98
GF Interval 8	0.99	0.99	0.99	0.99	0.98

Table 4.93. *AV1 Akiyo bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	238.64	166.52	111.59	77.48	56.38
GF Interval 16	339.41	248.44	175.63	132.71	97.60
GF Interval 8	344.41	254.67	183.90	133.60	97.76

GF has the same principle as I frame. It is fully intra coded frame and it is only a notation change for AV1; therefore, for sake of convenience in the graphs IF terms are used instead of GF. Figure 4.67 shows the behavior of AV1 for different I frame differences. I frame interval 16 and I frame interval 8 configurations results similar. As expected, I frame interval 8 configuration has the best quality outputs as well as highest bitrate values for all QP levels. The important aspect for AV1 is the decrease of the performance of the codec when the I frame interval increased from 16 to 24. The quality and bitrate drop are steep for the case I frame interval 24 configuration. Table 4.91, Table 4.92 and Table 4.93 supports the behavior as the I frame interval 24 configuration has more than 2 dB PSNR quality drop when the QP is selected 20.

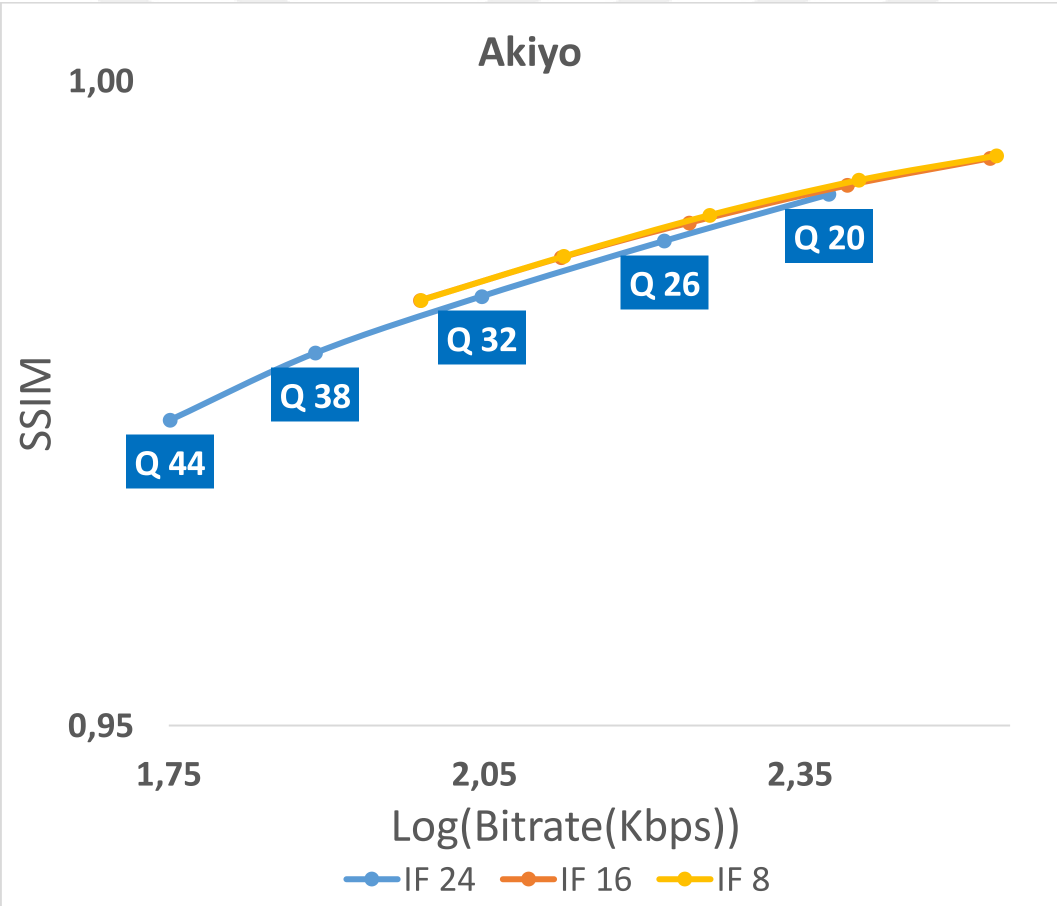

Figure 4.67. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.3.2. AV1 Bus

Table 4.94. *AV1 Bus PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	39.40	37.52	35.40	33.32	31.25
GF Interval 16	40.90	39.39	37.56	35.75	33.92
GF Interval 8	41.60	40.12	38.37	36.68	34.93

Table 4.95. *AV1 Bus SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	0.98	0.97	0.96	0.94	0.91
GF Interval 16	0.99	0.98	0.98	0.97	0.95
GF Interval 8	0.99	0.98	0.98	0.97	0.96

Table 4.96. *AV1 Bus bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	1678.62	1220.72	835.27	566.14	374.73
GF Interval 16	1968.82	1502.39	1092.19	793.04	572.85
GF Interval 8	2018.17	1553.99	1129.12	814.16	576.78

For video Bus the results are like Akiyo video. Figure 4.68 shows that for Bus video the difference between I frame interval 8 and I frame interval 16 configurations is more distinct. Moreover, I frame interval 24 configuration again performs worst in terms of quality outputs as well as results lowest bitrate values for all QP levels.

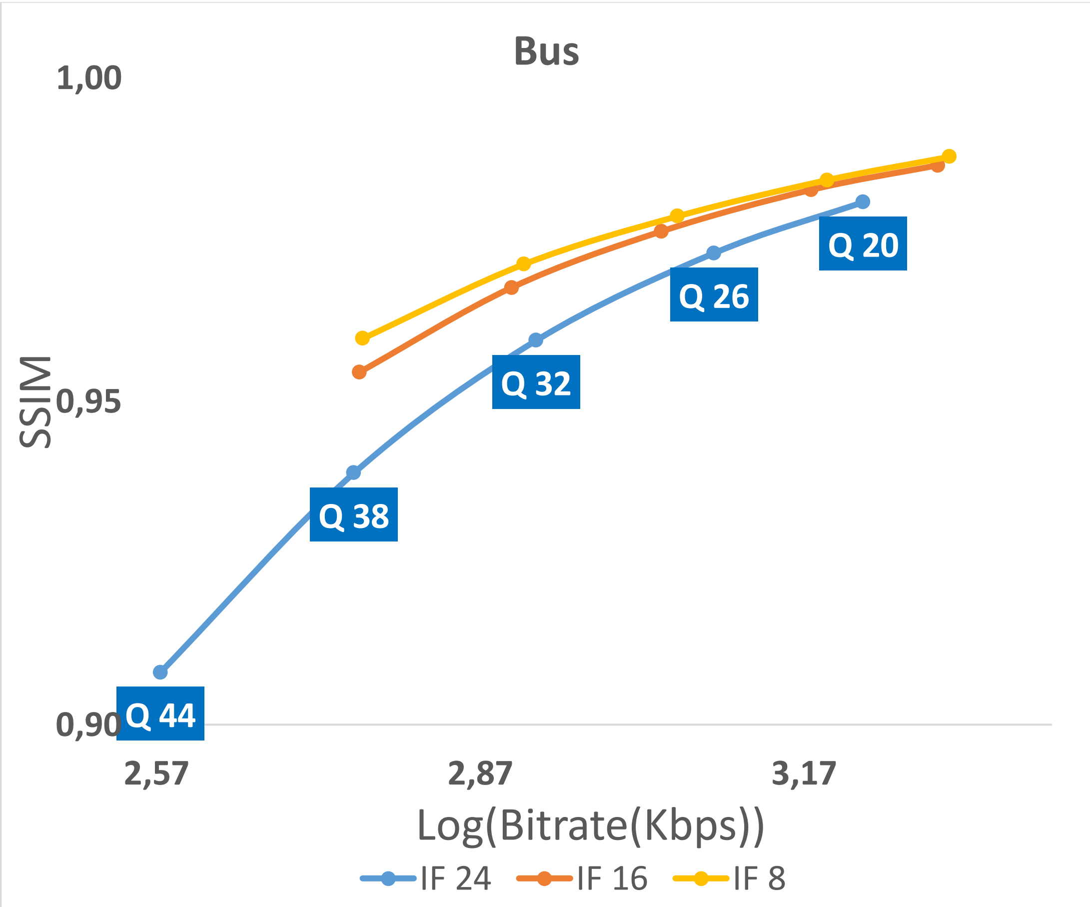

Figure 4.68. Bus SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.3.3. AV1 Container

Table 4.97. *AV1 Container PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	41.34	39.91	38.37	37.03	35.93
GF Interval 16	43.34	42.30	41.06	39.74	38.48
GF Interval 8	43.88	42.74	41.49	40.23	39.11

Table 4.98. *AV1 Container SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	0.97	0.95	0.94	0.92	0.91
GF Interval 16	0.98	0.97	0.96	0.95	0.94
GF Interval 8	0.98	0.98	0.97	0.96	0.95

Table 4.99. *AV1 Container bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	482.99	330.88	218.51	151.70	117.45
GF Interval 16	571.72	431.63	311.06	224.14	166.66
GF Interval 8	601.57	443.20	321.34	236.33	179.39

In Container video the results are similar to other CIF videos. I frame interval 8 configuration results best quality values as well as highest bitrate values.

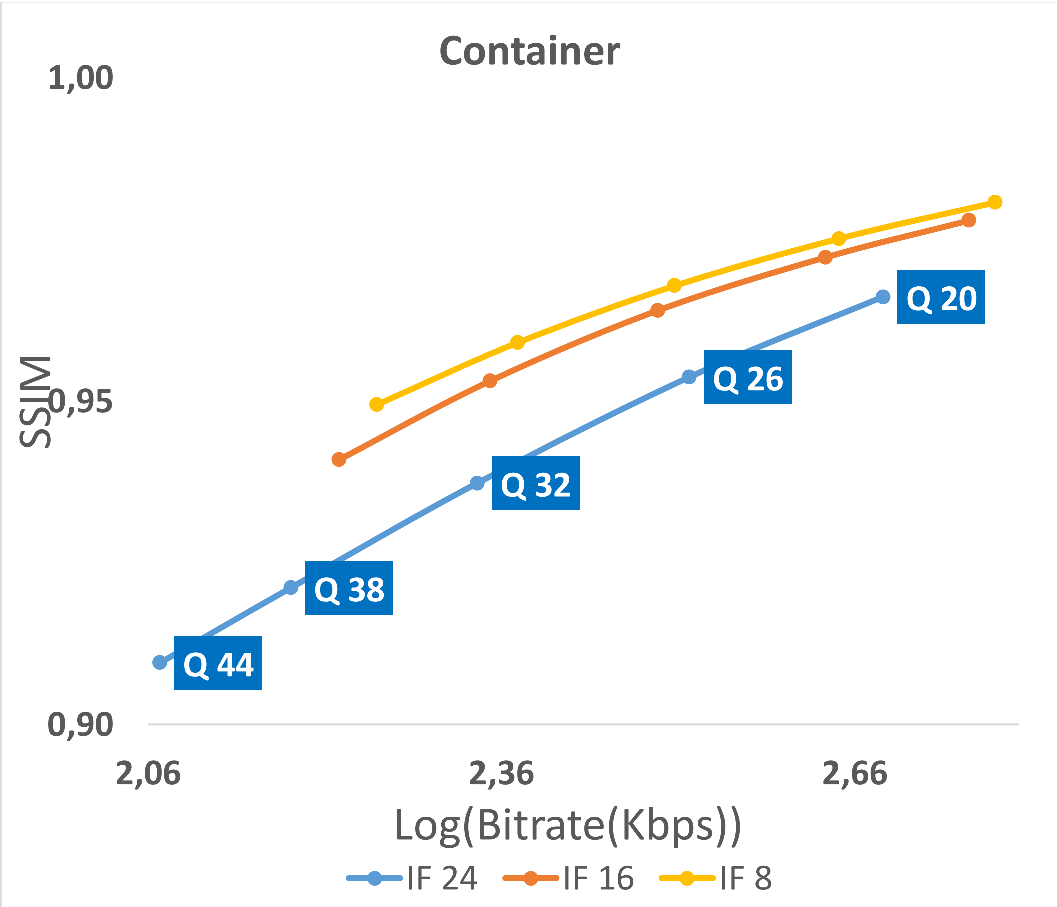

Figure 4.69. Container SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.3.4. AV1 Foreman

Table 4.100. *AV1 Foreman PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	41.43	40.20	38.68	37.11	35.50
GF Interval 16	42.52	41.49	40.27	39.03	37.77
GF Interval 8	42.92	41.89	40.69	39.54	38.34

Table 4.101. *AV1 Foreman SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	0.98	0.97	0.96	0.95	0.93
GF Interval 16	0.98	0.98	0.97	0.97	0.96
GF Interval 8	0.98	0.98	0.97	0.97	0.96

Table 4.102. *AV1 Foreman bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	668.74	479.74	319.71	216.64	145.10
GF Interval 16	848.01	638.59	455.63	334.94	245.03
GF Interval 8	889.33	665.74	471.11	340.84	247.75

Foreman video behaves similarly as other CIF videos. I frame interval 24 configuration has the worst quality outputs as well as lowest bitrate values.

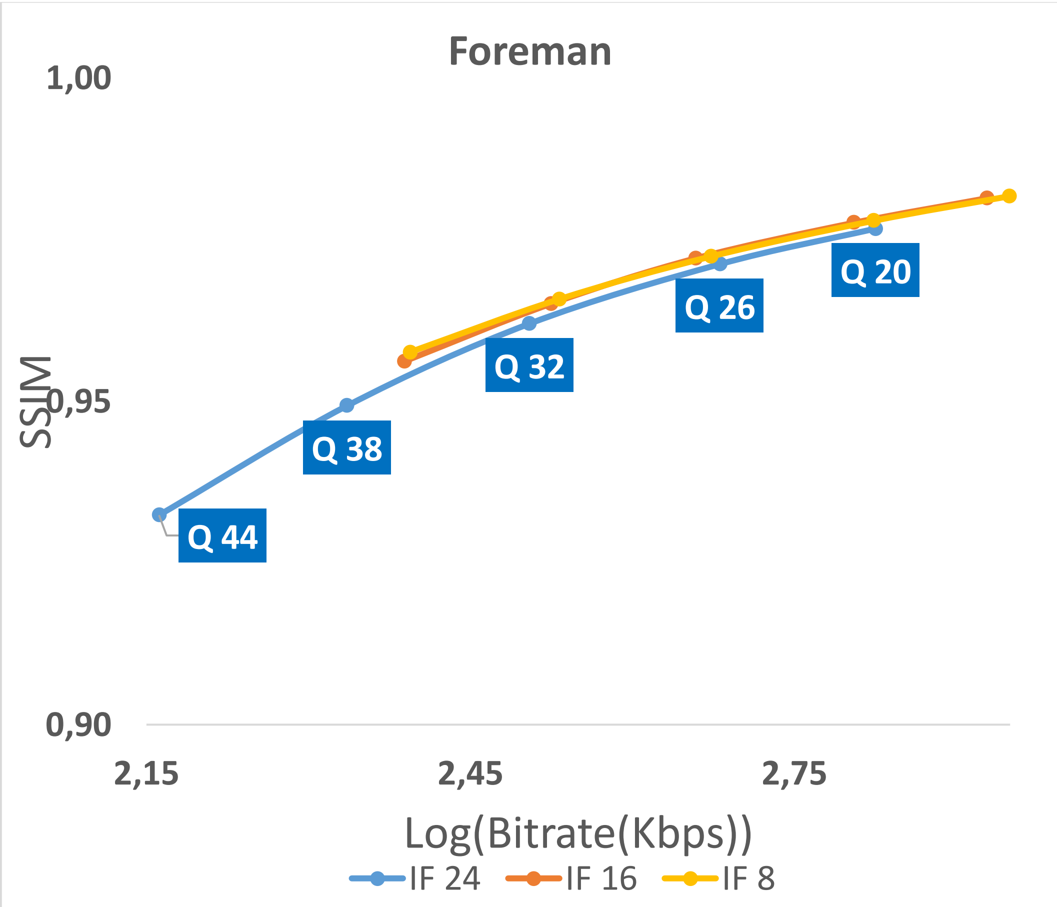

Figure 4.70. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.3.5. AV1 Mobile

Table 4.103. *AV1 Mobile PSNR values (dB) with different quantization values*

PSNR	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	39.36	37.16	34.74	32.56	30.67
GF Interval 16	40.60	38.81	36.83	35.02	33.43
GF Interval 8	41.16	39.31	37.37	35.53	33.91

Table 4.104. *AV1 Mobile SSIM values with different quantization values*

SSIM	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	0.99	0.99	0.98	0.97	0.96
GF Interval 16	0.99	0.99	0.99	0.98	0.98
GF Interval 8	0.99	0.99	0.99	0.98	0.98

Table 4.105. *AV1 Mobile bit rate values (kBps) with different quantization values*

Bit Rate	QP = 20	QP = 26	QP = 32	QP = 38	QP = 44
GF Interval 24	2468.43	1762.28	1125.24	698.30	438.30
GF Interval 16	2648.42	1994.94	1399.86	983.44	704.48
GF Interval 8	2824.28	2141.09	1532.95	1085.93	774.71

Mobile video shows also similar results with remaining CIF videos.

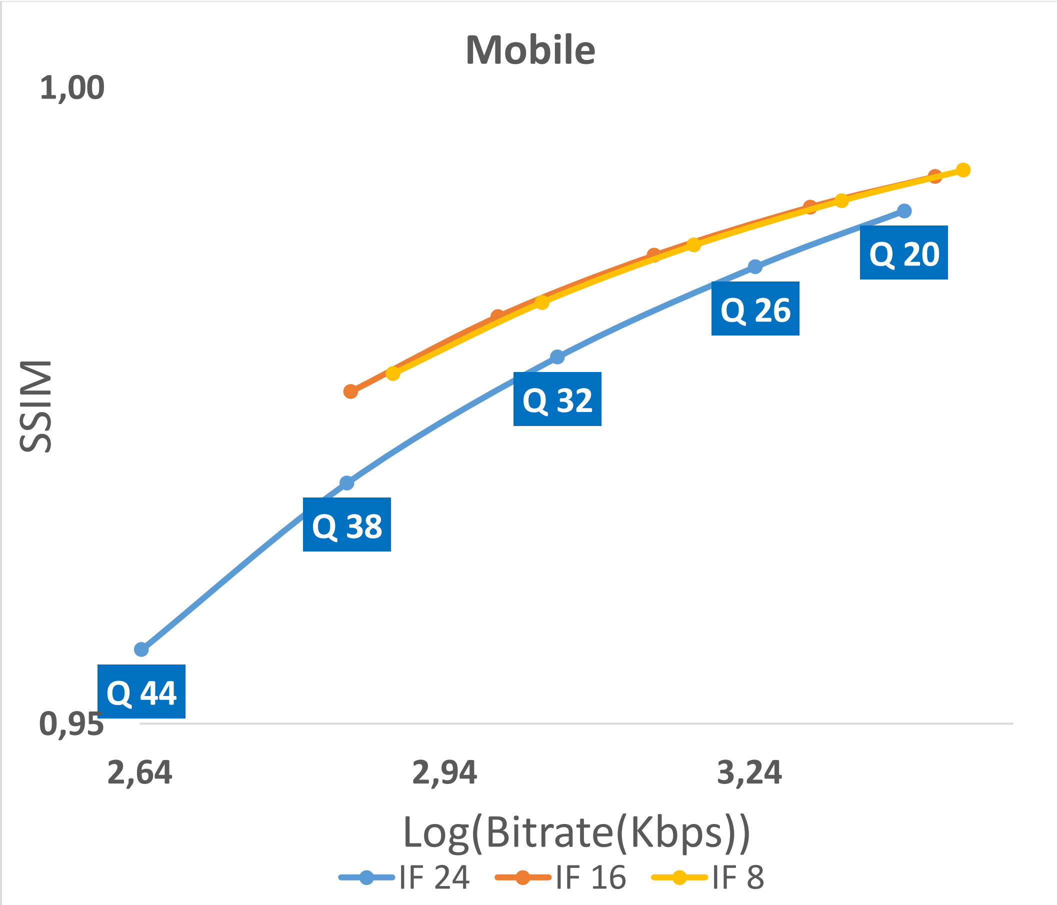

Figure 4.71. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different CTU and depth sizes

4.4.3.6. AV1 Basketball

Table 4.106. *AV1 Basketball PSNR values (dB) with different quantization values*

PSNR	**QP = 20**	**QP = 26**	**QP = 32**	**QP = 38**	**QP = 44**
GF Interval 24	47.80	47.21	46.35	45.26	43.97
GF Interval 16	48.24	47.80	47.20	46.48	45.57
GF Interval 8	48.24	47.80	47.20	46.49	45.60

Table 4.107. *AV1 Basketball SSIM values with different quantization values*

SSIM	**QP = 20**	**QP = 26**	**QP = 32**	**QP = 38**	**QP = 44**
GF Interval 24	0.99	0.99	0.99	0.99	0.98
GF Interval 16	0.99	0.99	0.99	0.99	0.99
GF Interval 8	0.99	0.99	0.99	0.99	0.99

Table 4.108. *AV1 Basketball bit rate values (kBps) with different quantization values*

Bit Rate	**QP = 20**	**QP = 26**	**QP = 32**	**QP = 38**	**QP = 44**
GF Interval 24	7591.19	5147.64	3529.35	2464.44	1767.29
GF Interval 16	10033.10	6839.27	4686.37	3328.69	2410.30
GF Interval 8	10582.05	7131.20	4848.17	3461.16	2497.16

In video Basketball the difference between I frame interval 16 and I frame interval 8 configurations is diminished. For lower levels of QP, the quality levels are same while I frame interval 8 configuration results higher bitrate values. However, for high levels of QP I frame interval 8 configuration has slight improvement over I frame interval 16 configuration in terms of quality. This behavior is different from CIF videos, in CIF videos between I frame interval 8 and I frame interval 16 configurations have significant difference for all QP levels, while for 4K video the importance of having more frequent I frame than 16 is not that affective for AV1 codec.

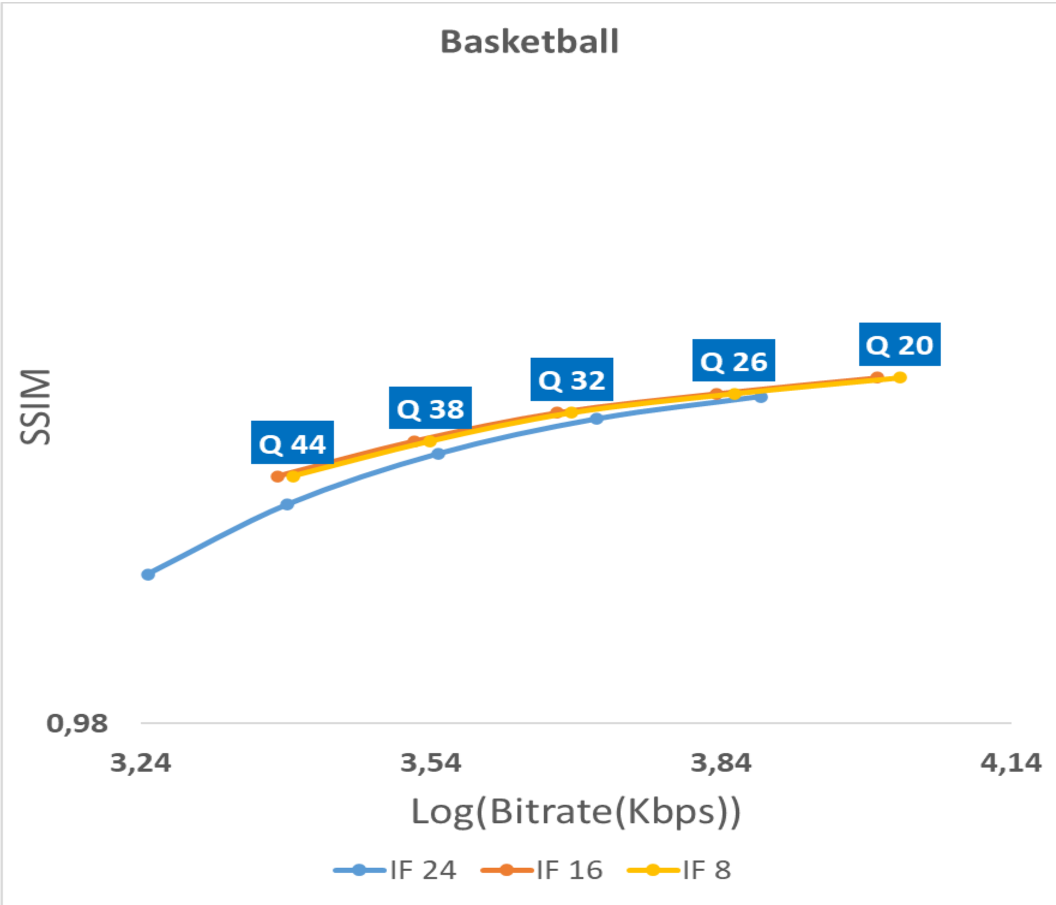

Figure 4.72. Basketball SSIM vs Log(Bitrate(Kbps) graph for different CTU and depth sizes

4.4.3.7. AV1 BD Rate results

For AV1 the results demonstrate that increasing I frame interval results with worse quality values as well as lower bitrate values. I frame interval 8 and I frame interval 16 configurations show closer outputs while increasing I frame interval to 24 AV1 starts to perform worse than other 2 configurations. For CIF videos best results are acquired when the I frame interval is chosen as 8. It gives the best quality levels as well as highest bitrate values. From Figure 4.73 it can be seen that choosing lower I frame interval is more efficient than other configuration especially for more complex videos such as Bus.

For 4K Basketball video the results are different. I frame interval 8 configuration has the best quality output as well as highest bitrate values. However, the efficiency of I frame interval 8 configuration is worse than I frame interval 16 configuration. Since bitrate increase for I frame interval 8 configuration does not result sufficient quality increase.

When the I frame interval 8 and I frame interval 16 configurations are compared, for CIF videos I frame interval 8 configuration always perform better than the I frame interval 16 configuration; however, for 4K video Basketball I frame interval 16 performs better than I frame interval 8. This is because of the resolution of the video, creating an I frame in every 8 frames creates substantial bitrate increase for 4K resolution since there is no inter predictions performed for encoding process; therefore even though I frame interval 8 gives better quality result as it can be seen from Figure 4.72 the bitrate increase is too much for higher resolution videos. However, for lower resolution CIF videos creating I frame with interval 8 does not bring that much bitrate increase to the encoding process. Therefore, choosing a decent I frame interval is important for AV1 codec when especially higher resolution videos are used, for lower resolution videos choosing smaller I frame interval makes the results better overall.

			IF 8	IF 16	IF 24
Akiyo	IF 8	BDBR		12,44	25,59
		BDPSNR		-0,48	-0,93
	IF 16	BDBR	-11,06		11,06
		BDPSNR	0,48		-0,43
	IF 24	BDBR	-20,38	-9,96	
		BDPSNR	0,93	0,43	
			IF 8	IF 16	IF 24
Bus	IF 8	BDBR		19,81	31,59
		BDPSNR		-0,99	-1,50
	IF 16	BDBR	-16,53		9,18
		BDPSNR	0,99		-0,49
	IF 24	BDBR	-24,01	-8,41	
		BDPSNR	1,50	0,49	
			IF 8	IF 16	IF 24
Container	IF 8	BDBR		8,20	52,59
		BDPSNR		-0,31	-1,45
	IF 16	BDBR	-7,58		40,95
		BDPSNR	0,31		-1,05
	IF 24	BDBR	-34,46	-29,05	
		BDPSNR	1,45	1,05	
			IF 8	IF 16	IF 24
Foreman	IF 8	BDBR		16,25	19,71
		BDPSNR		-0,56	-0,67
	IF 16	BDBR	-13,98		3,09
		BDPSNR	0,56		-0,11
	IF 24	BDBR	-16,46	-2,99	
		BDPSNR	0,67	0,11	
			IF 8	IF 16	IF 24
Mobile	IF 8	BDBR		0,95	19,40
		BDPSNR		-0,05	-0,96
	IF 16	BDBR	-0,94		18,64
		BDPSNR	0,05		-0,91
	IF 24	BDBR	-16,25	-15,71	
		BDPSNR	0,96	0,91	
			IF 8	IF 16	IF 24
Basketball	IF 8	BDBR		-3,63	6,98
		BDPSNR		0,07	-0,15
	IF 16	BDBR	3,77		10,68
		BDPSNR	-0,07		-0,23
	IF 24	BDBR	-6,53	-9,65	
		BDPSNR	0,15	0,23	

Figure 4.73. AV1 BD Rates of all videos for different CTU and depth sizes

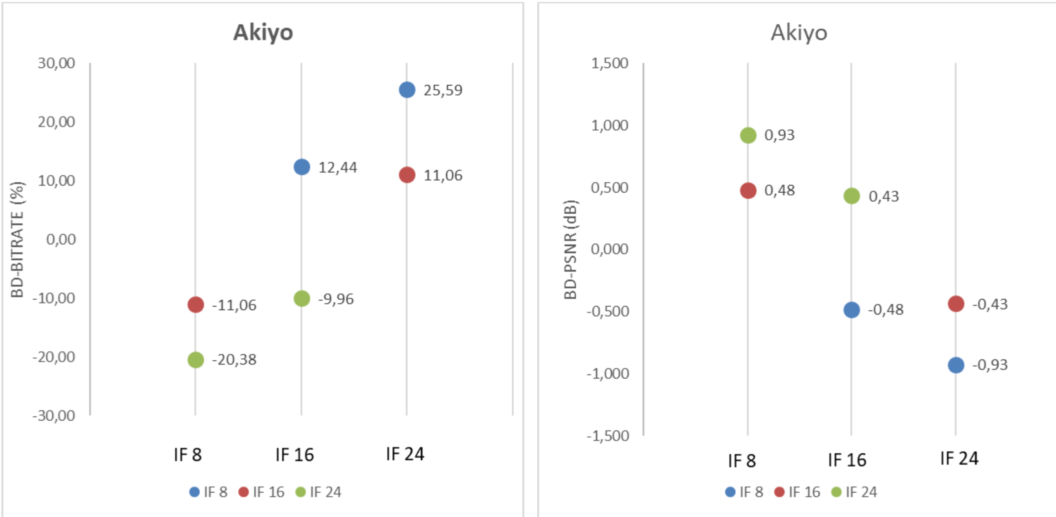

Figure 4.74. Akiyo BD Rates for AV1

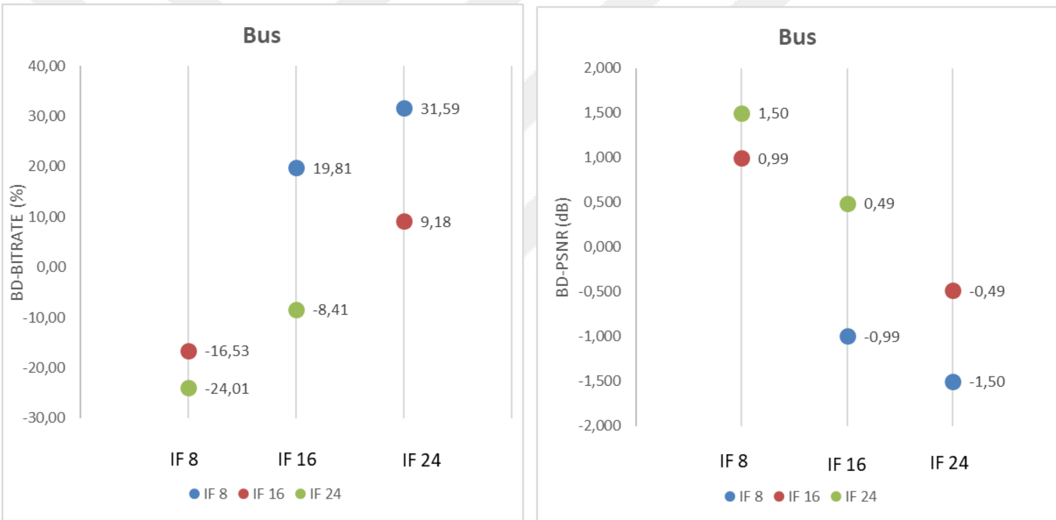

Figure 4.75. Bus BD Rates for AV1

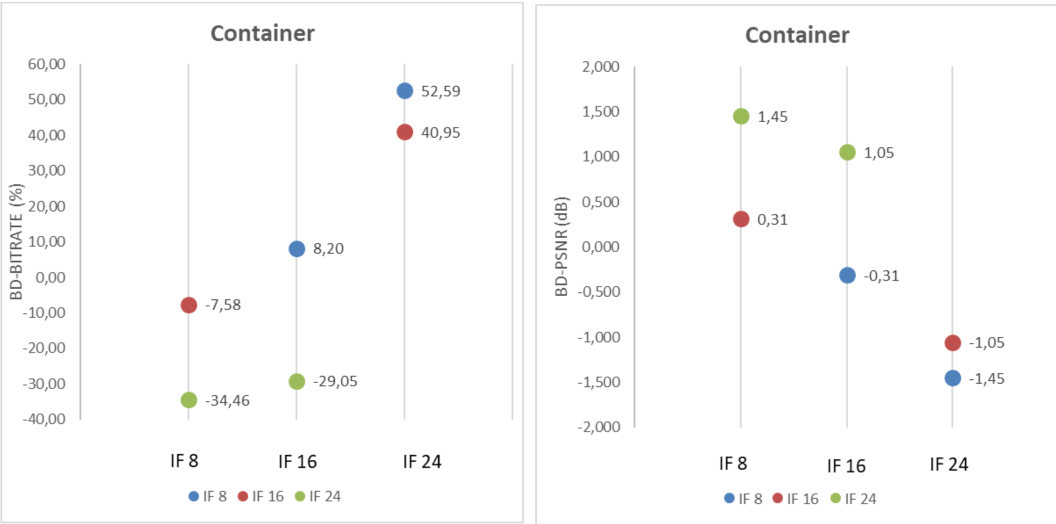

Figure 4.76. Container BD Rates for AV1

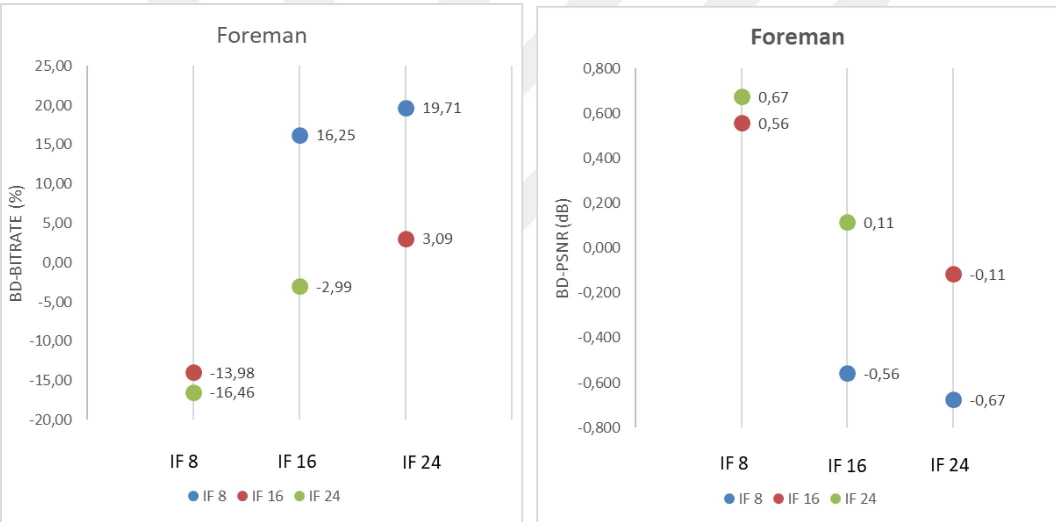

Figure 4.77. Foreman BD Rates for AV1

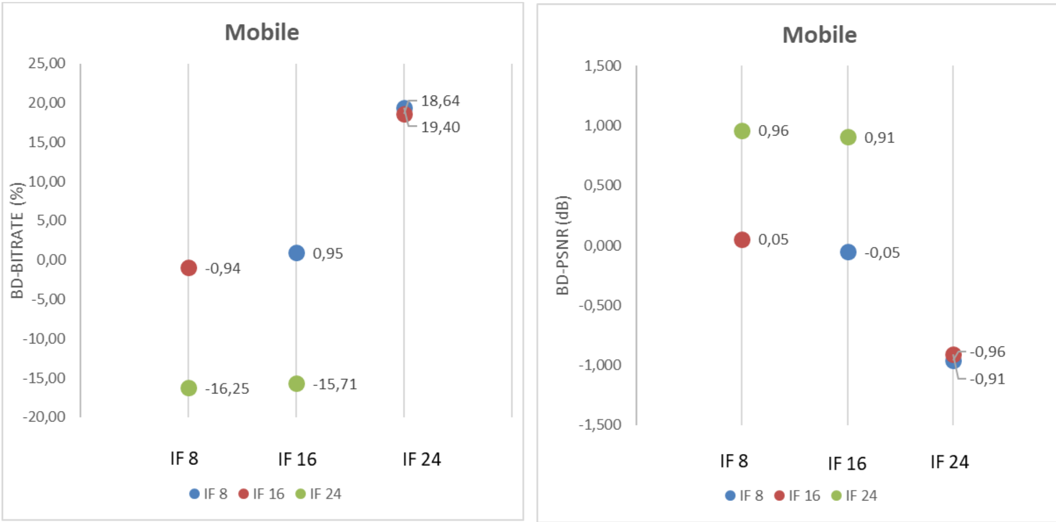

Figure 4.78. Mobile BD Rates for AV1

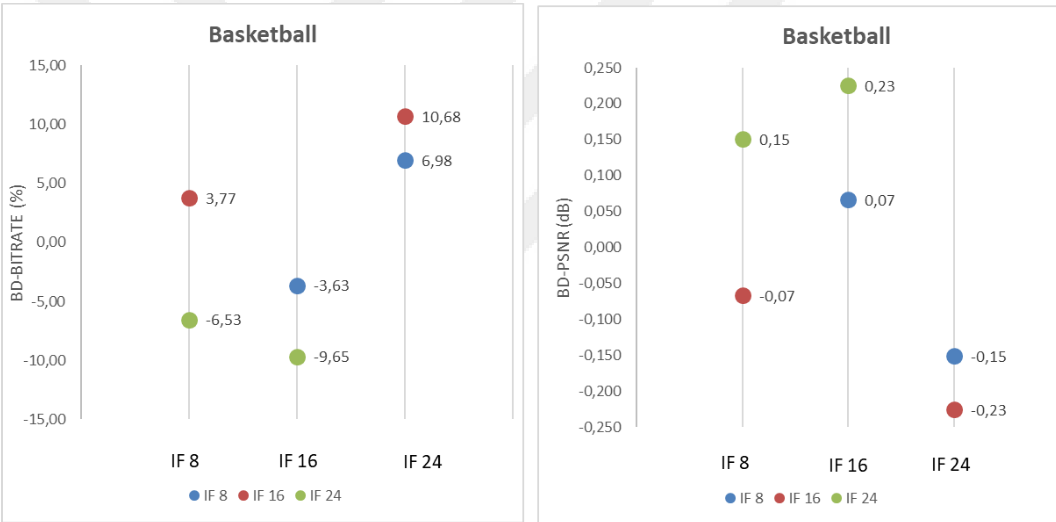

Figure 4.79. *Basketball BD Rates for AV1*

4.5. Bit Per Second (BPS) Configuration

In this step codecs are compared with each other directly. Codecs are configured for rate control, rather than quality control which takes into consideration of quantization. For CIF videos bps values are selected as 100 kBps, 200 kBps, 500 kBps and 1000 kBps. For 4K video the bps values should be selected higher since higher the resolution needs more bit rates to perform encoding. Therefore, bps values are selected as 1 Mbps, 1.5 Mbps, 2Mbps and 2.5 Mbps. First 150 frames are processed for all videos, frame rate is 30 and internal bit depth is 8.

4.5.1. BPS Akiyo

Table 4.109. *Akiyo PSNR values (dB) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	42.96	44.85	47.23	49.33
AV1	43.44	46.16	49.57	52.27
HEVC	41.76	44.15	46.89	49.55

Table 4.110. *Akiyo SSIM values for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	0.98	0.99	0.99	0.99
AV1	0.98	0.99	1.00	0.99
HEVC	0.98	0.98	0.99	0.99

Table 4.111. *Akiyo bit rate values (kBps) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	101.25	201.47	501.72	1002.41
AV1	101.49	196.54	456.57	929.47
HEVC	102.19	201.30	501.18	1001.38

For Akiyo video, Figure 4.80 shows AV1 performs better than VVC and VVC performs better than HEVC in terms of quality at the same bitrate level. AV1 performs better in Video Akiyo because Akiyo has the lowest complexity in terms of both spatial information and temporal information. Table 4.109, Table 4.110 and Table 4.111 gives the numerical outputs and AV1 is better more than 2 dB for higher bitrate values than VVC.

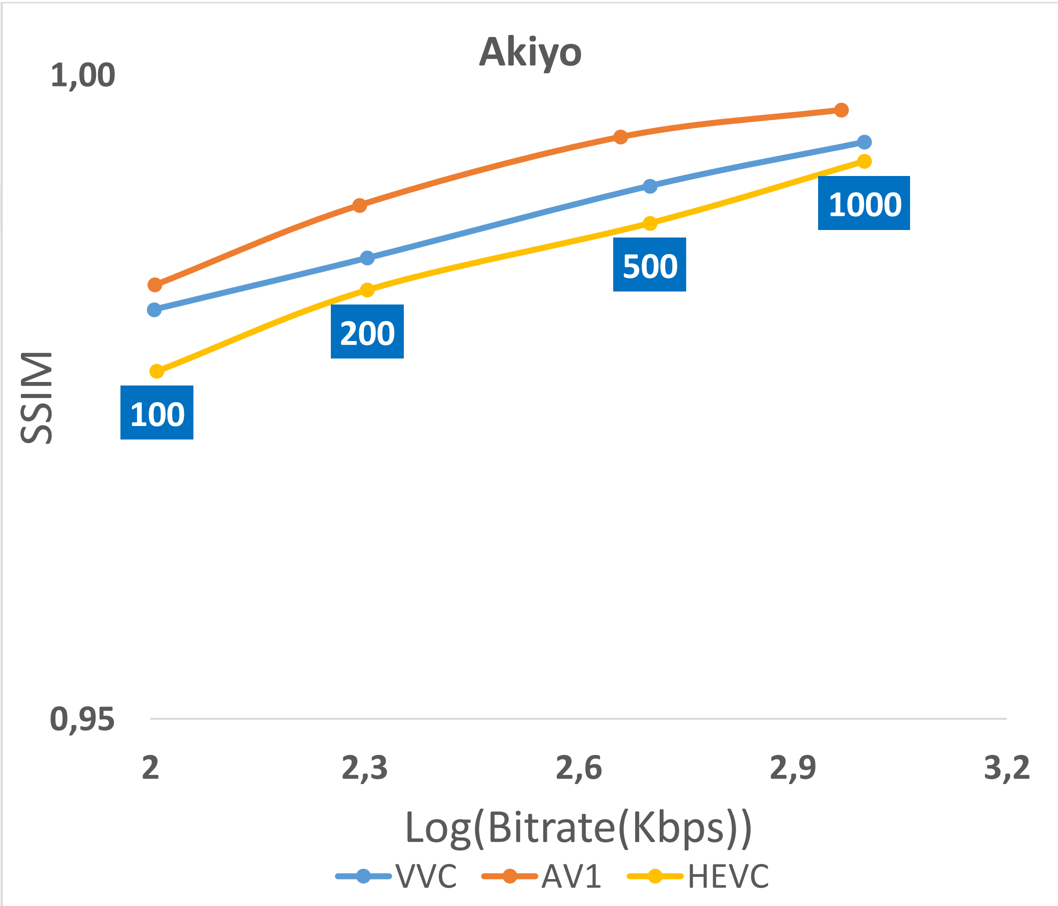

Figure 4.80. Akiyo SSIM vs Log(Bitrate(Kbps)) graph for different BPS values

4.5.2. BPS Bus

Table 4.112. *Bus PSNR values (dB) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	27.07	30.23	34.83	38.39
AV1	26.77	29.49	33.69	37.12
HEVC	25.33	28.35	32.88	36.54

Table 4.113. *Bus SSIM values for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	0.82	0.90	0.96	0.98
AV1	0.79	0.87	0.95	0.97
HEVC	0.76	0.86	0.94	0.97

Table 4.114. *Bus bit rate values (kBps) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	101.28	201.58	501.72	1002.12
AV1	107.40	190.91	442.72	840.05
HEVC	101.73	201.10	501.19	1001.54

When the Bus video is examined from Figure 4.81 VVC outperforms AV1 and AV1 outperforms HEVC. VVC performs better than AV1 for video Bus because the complexity of video Bus is higher than video Akiyo. It can be inferred that VVC performs better than AV1 when the complexity of the video gets higher.

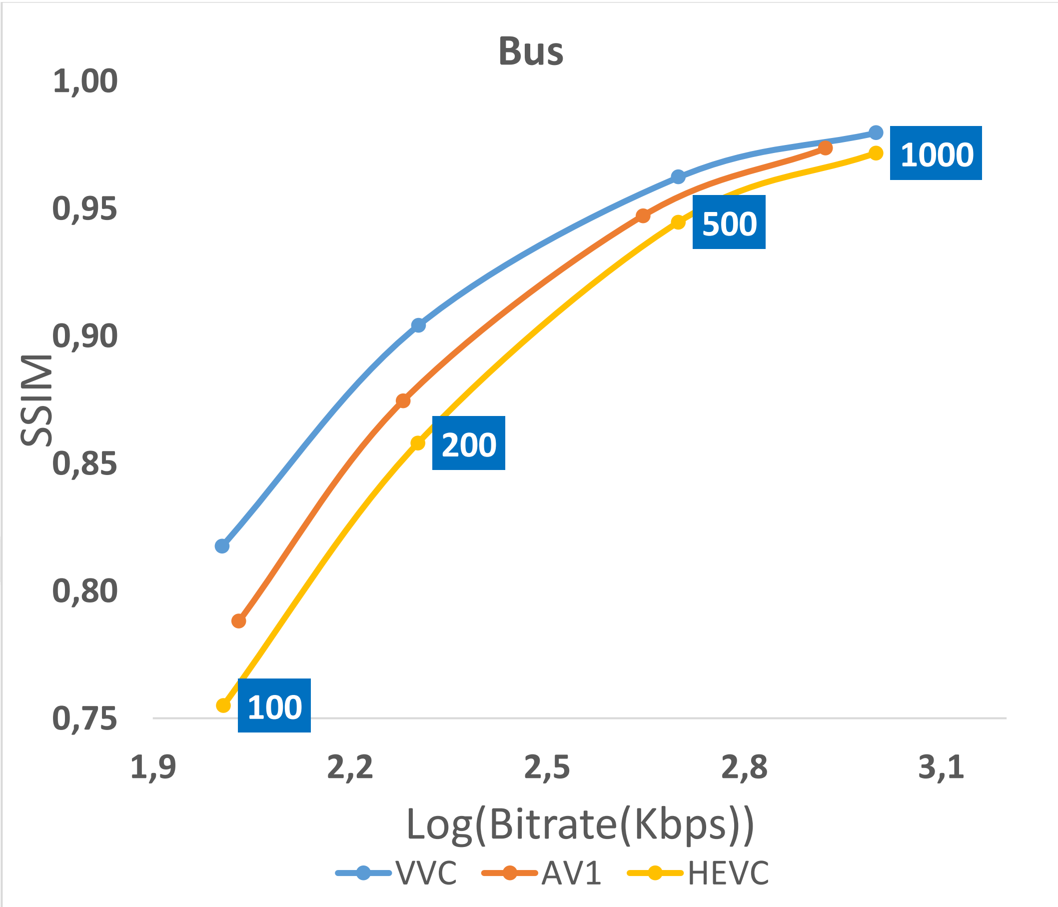

Figure 4.81. Bus SSIM vs Log(Bitrate(Kbps)) graph for different BPS values

4.5.3. BPS Container

Table 4.115. *Container PSNR values (dB) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	36.81	39.20	42.89	45.37
AV1	36.86	39.74	43.48	46.08
HEVC	35.27	37.85	41.42	44.44

Table 4.116. *Container SSIM values for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	0.93	0.95	0.98	0.99
AV1	0.92	0.95	0.98	0.99
HEVC	0.91	0.94	0.97	0.98

Table 4.117. *Container bit rate values (kBps) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	102.54	201.44	495.88	1008.94
AV1	107.70	219.06	582.10	1160.65
HEVC	103.63	201.18	501.41	1002.07

For video Container, Figure 4.81 indicates that VVC and AV1 performs similar while HEVC performs worse than others. AV1's performance is near to VVC's for video Container because its temporal information is low as Akiyo; however, it has higher spatial information and higher complexity. It is enough for VVC to perform better than AV1 for a video having higher spatial information.

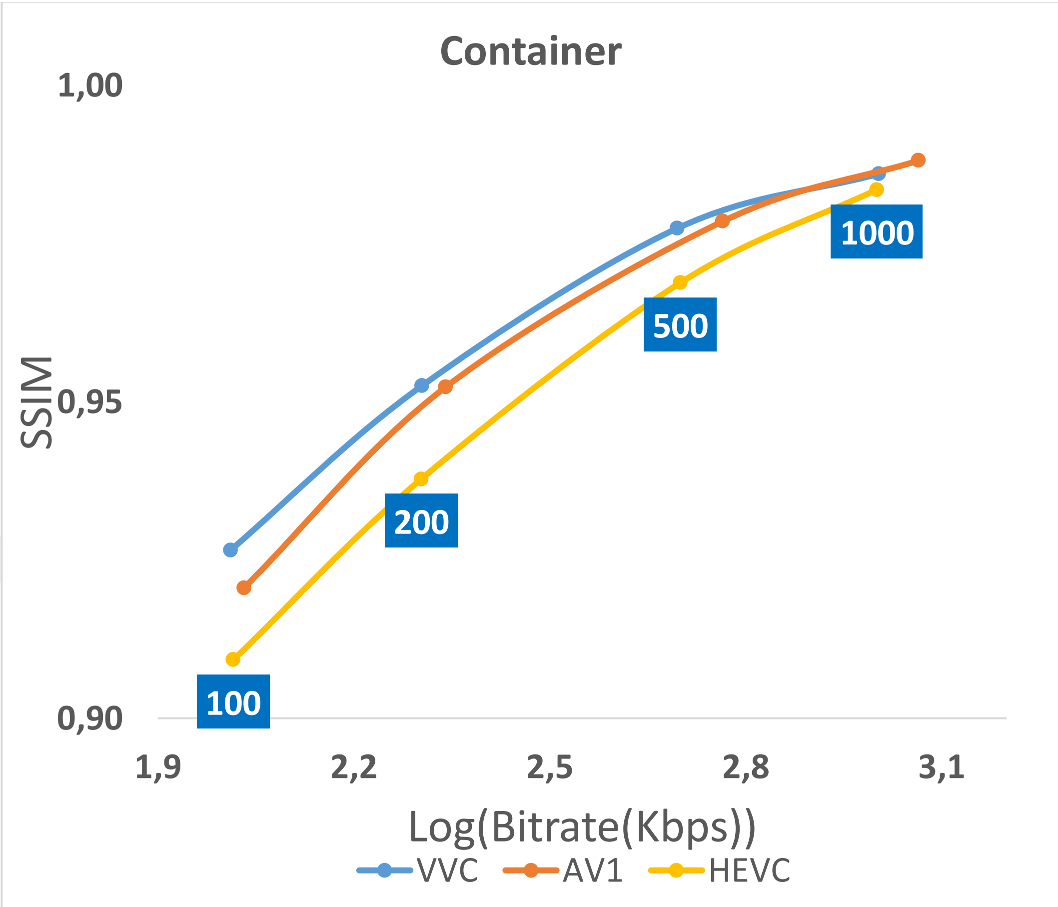

Figure 4.82. Container SSIM vs Log(Bitrate(Kbps)) graph for different BPS values

4.5.4. BPS Foreman

Table 4.118. *Foreman PSNR values (dB) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	34.94	37.62	41.19	43.55
AV1	35.16	37.83	41.43	43.85
HEVC	33.49	36.17	40.15	42.72

Table 4.119. *Foreman SSIM values for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	0.92	0.95	0.98	0.98
AV1	0.93	0.95	0.98	0.99
HEVC	0.90	0.94	0.97	0.98

Table 4.120. *Foreman bit rate values (kBps) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	101.73	200.89	495.19	1002.41
AV1	110.56	213.03	515.69	1032.59
HEVC	105.02	201.82	501.24	1001.24

For video Foreman, since it has lower spatial information, AV1 codec competes with VVC, in fact it results better quality values for desired bitrate specifications, as it can be seen from Table 4.118, Table 4.119 and Table 4.120. HEVC again performs worse than both AV1 and VVC.

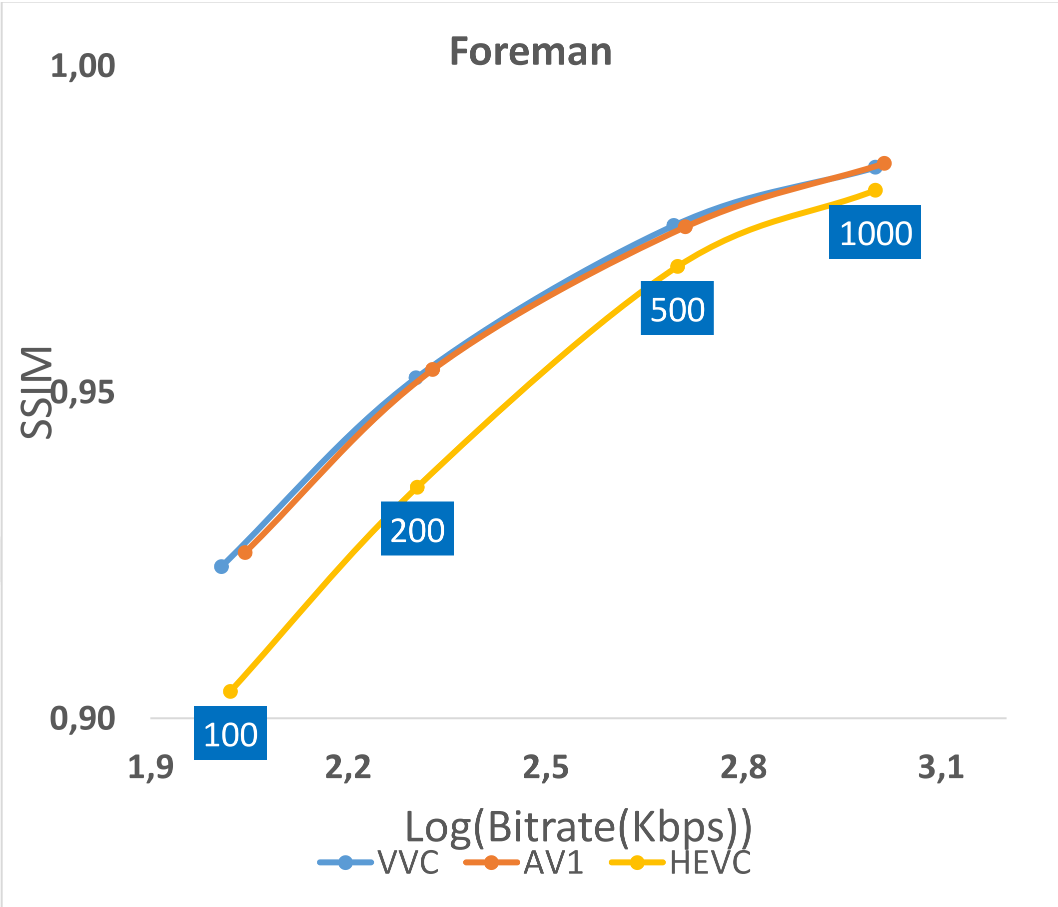

Figure 4.83. Foreman SSIM vs Log(Bitrate(Kbps)) graph for different BPS values

4.5.5. BPS Mobile

Table 4.121. *Mobile PSNR values (dB) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	26.46	29.36	32.99	36.05
AV1	26.42	28.60	32.51	35.51
HEVC	24.16	27.53	31.29	34.54

Table 4.122. *Mobile SSIM values for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	0.90	0.95	0.97	0.98
AV1	0.89	0.93	0.97	0.98
HEVC	0.82	0.91	0.96	0.98

Table 4.123. *Mobile bit rate values (kBps) for different BPS configurations*

PSNR	100 kBps	200 kBps	500 kBps	1000 kBps
VVC	109.36	210.22	502.25	1001.57
AV1	129.69	204.26	478.48	909.85
HEVC	109.04	212.04	501.42	1001.18

For a complex video such as Mobile VVC starts to perform better than AV1 especially for lower bitrate levels. Figure 4.84 shows the behavior of AV1 and it catches VVC in terms of quality when the bitrate value increased to 1000 kBps. HEVC performs worse than AV1 and VVC as in all other CIF videos.

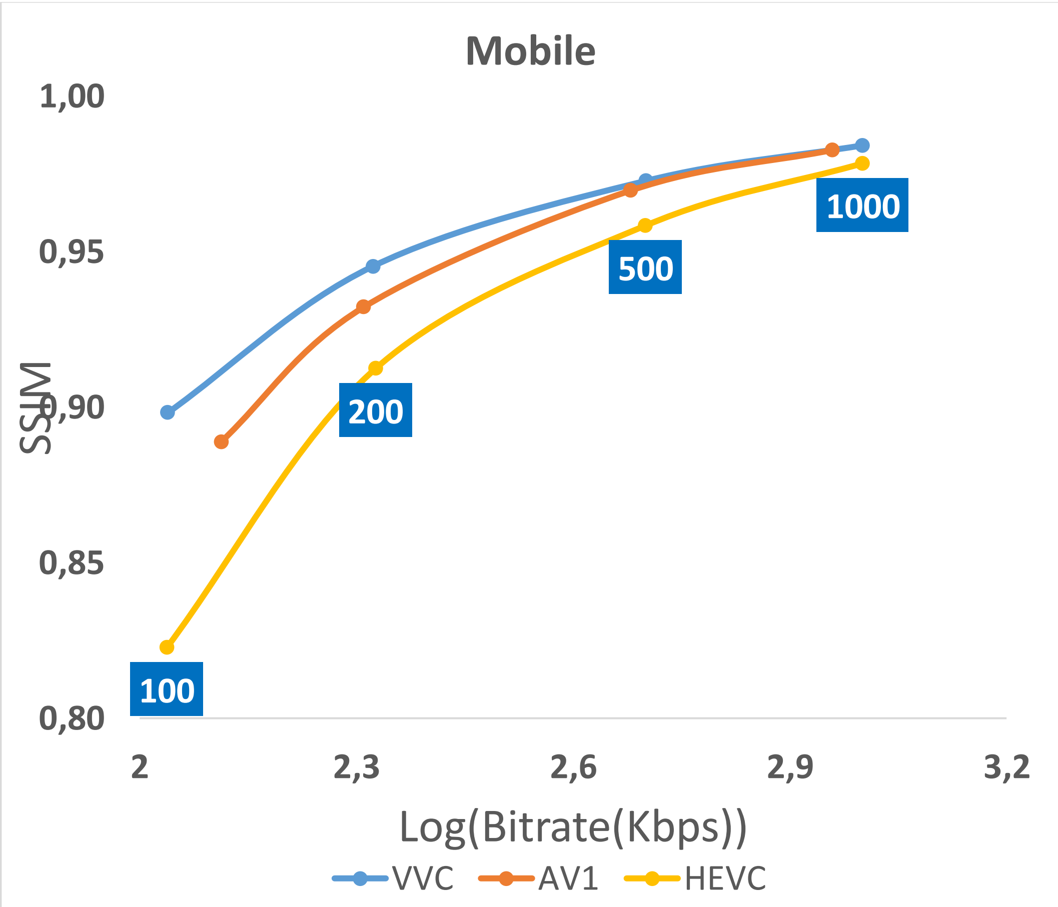

Figure 4.84. Mobile SSIM vs Log(Bitrate(Kbps)) graph for different BPS values

4.5.6. BPS Basketball

Table 4.124. *Basketball PSNR values (dB) for different BPS configurations*

PSNR	1Mbps	1.5Mbps	2Mbps	2.5Mbps
VVC	42.30	44.18	45.25	45.91
AV1	41.89	43.77	44.83	45.36
HEVC	39.94	42.20	43.56	44.46

Table 4.125. *Basketball SSIM values for different BPS configurations*

PSNR	1Mbps	1.5Mbps	2Mbps	2.5Mbps
VVC	0.98	0.99	0.99	0.99
AV1	0.98	0.99	0.99	0.99
HEVC	0.98	0.98	0.98	0.99

Table 4.126. *Basketball bit rate values (kBps) for different BPS configurations*

PSNR	1Mbps	1.5Mbps	2Mbps	2.5Mbps
VVC	1001.81	1504.75	2006.04	2501.25
AV1	1042.04	1540.12	1995.12	2305.98
HEVC	1001.68	1501.17	2001.20	2501.26

In 4K video Basketball, VVC performs better than HEVC and AV1 for all bitrate levels. Figure 4.85 shows the difference between codecs for different bitrate levels. Bitrate levels are chosen different from CIF videos, since encoding a 4K video at 100 kBps is illogical, since the data need for 4K video is lot more than CIF videos. The gap between AV1 and HEVC is more than the GAP between VVC and AV1. VVC and AV1 are the future of the video codecs and they are encoding more efficiently than HEVC, especially for higher quality videos.

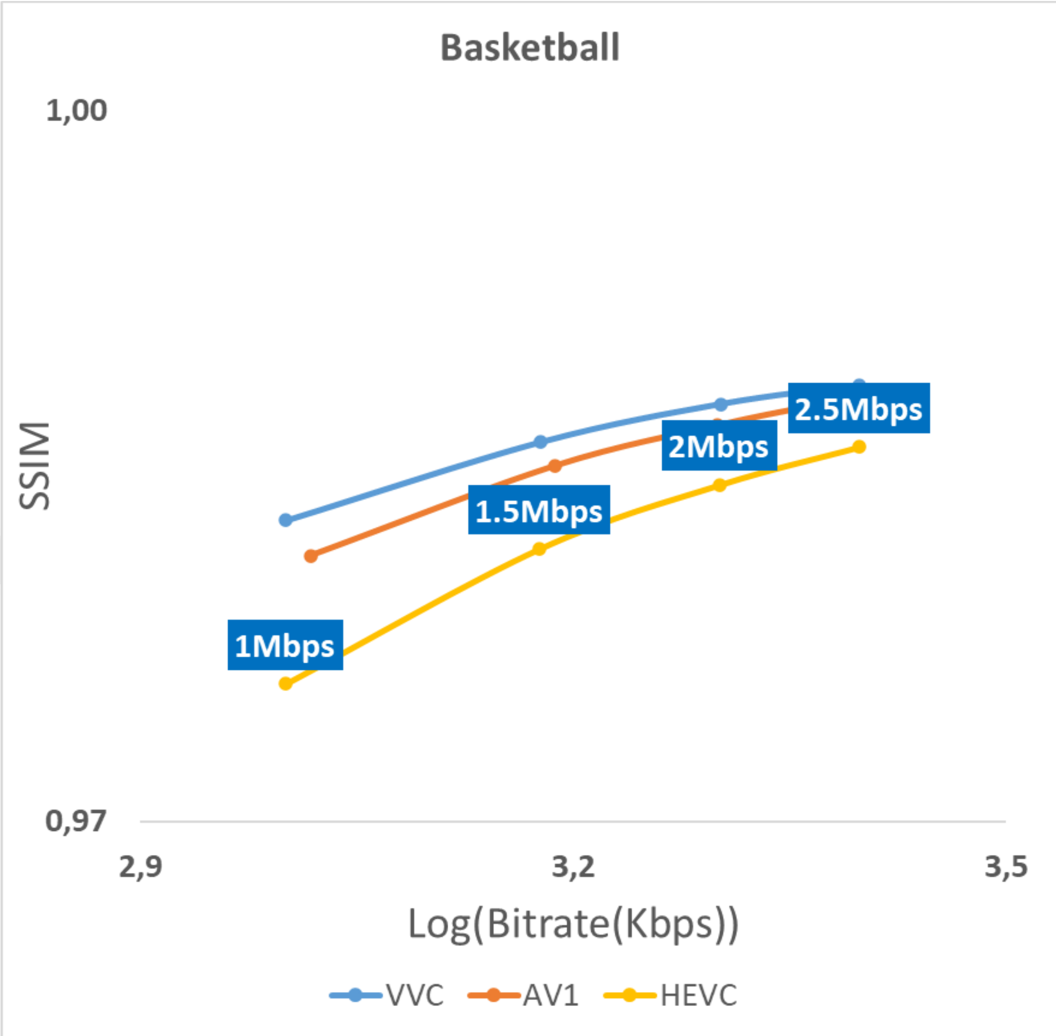

Figure 4.85. Basketball PSNR vs Log(Bitrate(Kbps) graph for different BPS values

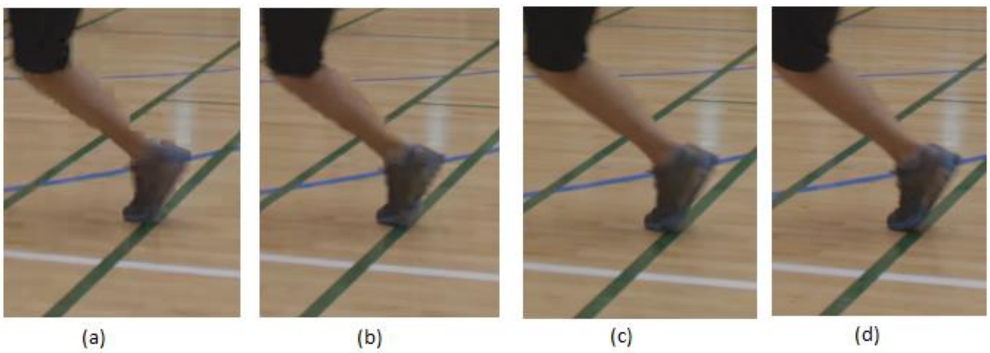

Figure 4.86. Comparison of codecs at 1Mbps for basketball, a: HEVC, b: AV1, c: AV1, d: original

4.5.7. BPS BD Rate results

At the same BPS rate, VVC outperforms both AV1 and HEVC. For lower resolution videos when the complexity is low AV1 can perform better than VVC; however, for complex videos in lower resolutions VVC outperforms AV1. For 4K video the VVC clearly better than AV1 and AV1 is better than HEVC.

For 4K video Basketball VVC is better than other codecs, saves %10,90 bitrate when compared to AV1 and saves %34,63 bitrate when compared to HEVC at the same quality level. Moreover, AV1 results 0,49 dB worse PSNR when compared to VVC and 1,44 dB better PSNR when compared to HEVC at the same bitrate level. AV1 tries to keep up with VVC; however, HEVC drop far behind of these two new codecs.

Figure 4.86 illustrates the difference between codecs at the 33rd frame showing the number 4 players right leg. This leg has the biggest and fastest movement in the video right that frame and hardest part to encode. It is clearly seen that how bad HEVC decoded that leg area while AV1 encoded better than HEVC, still can be detected by human eye easily. VVC clearly is the winner and its result is almost same as the original video, without looking careful the small deficiencies cannot be detected.

			AV1-BPS	HEVC-BPS	VVC-BPS
Akiyo	AV1-BPS	BDBR		100,09	72,61
		BDPSNR		-2,52	-1,98
	HEVC-BPS	BDBR	-50,02		-14,46
		BDPSNR	2,52		0,51
	VVC-BPS	BDBR	-42,07	16,91	
		BDPSNR	1,98	-0,51	
			AV1-BPS	HEVC-BPS	VVC-BPS
Bus	AV1-BPS	BDBR		32,84	-9,27
		BDPSNR		-1,41	0,48
	HEVC-BPS	BDBR	-24,72		-31,79
		BDPSNR	1,41		1,89
	VVC-BPS	BDBR	10,22	46,61	
		BDPSNR	-0,48	-1,89	
			AV1-BPS	HEVC-BPS	VVC-BPS
Container	AV1-BPS	BDBR		45,94	2,23
		BDPSNR		-1,47	-0,09
	HEVC-BPS	BDBR	-31,48		-30,00
		BDPSNR	1,47		1,39
	VVC-BPS	BDBR	-2,19	42,86	
		BDPSNR	0,09	-1,39	
			AV1-BPS	HEVC-BPS	VVC-BPS
Foreman	AV1-BPS	BDBR		37,24	0,82
		BDPSNR		-1,28	-0,03
	HEVC-BPS	BDBR	-27,13		-27,07
		BDPSNR	1,28		1,26
	VVC-BPS	BDBR	-0,81	37,11	
		BDPSNR	0,03	-1,26	
			AV1-BPS	HEVC-BPS	VVC-BPS
Mobile	AV1-BPS	BDBR		34,43	-9,44
		BDPSNR		-1,36	0,43
	HEVC-BPS	BDBR	-25,61		-33,26
		BDPSNR	1,36		1,80
	VVC-BPS	BDBR	10,43	49,83	
		BDPSNR	-0,43	-1,80	
			AV1-BPS	HEVC-BPS	VVC-BPS
Basketbal	AV1-BPS	BDBR		36,17	-10,90
		BDPSNR		-1,44	0,49
	HEVC-BPS	BDBR	-26,56		-34,63
		BDPSNR	1,44		1,91
	VVC-BPS	BDBR	12,23	52,98	
		BDPSNR	-0,49	-1,91	

Figure 4.87. BD Rates of all videos for different bps configurations

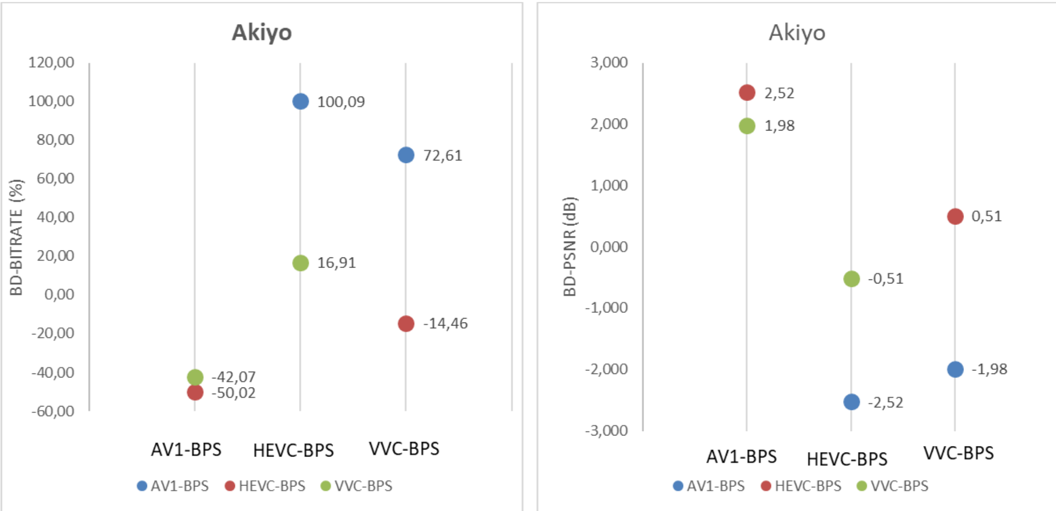

Figure 4.88. Akiyo BD Rates

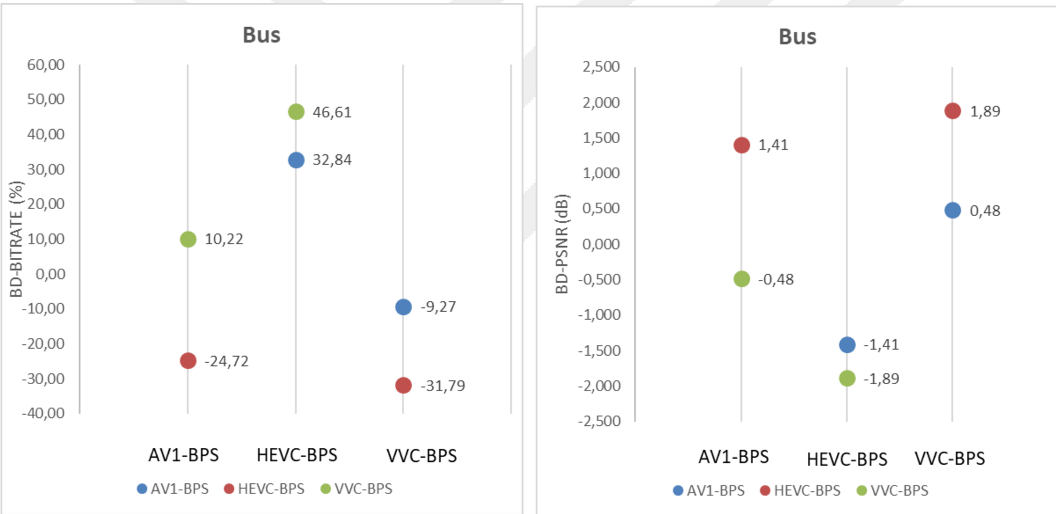

Figure 4.89. Bus BD Rates

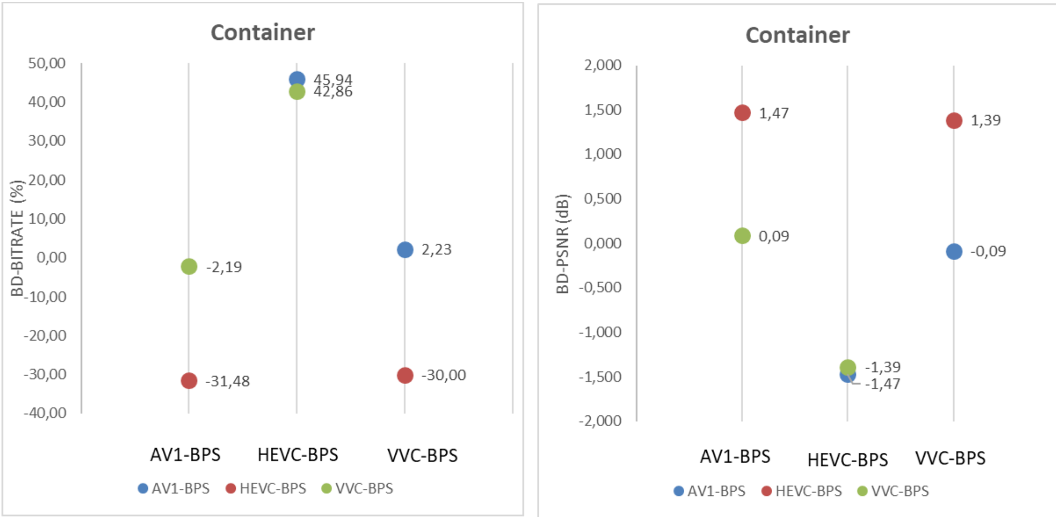

Figure 4.90. Container BD Rates

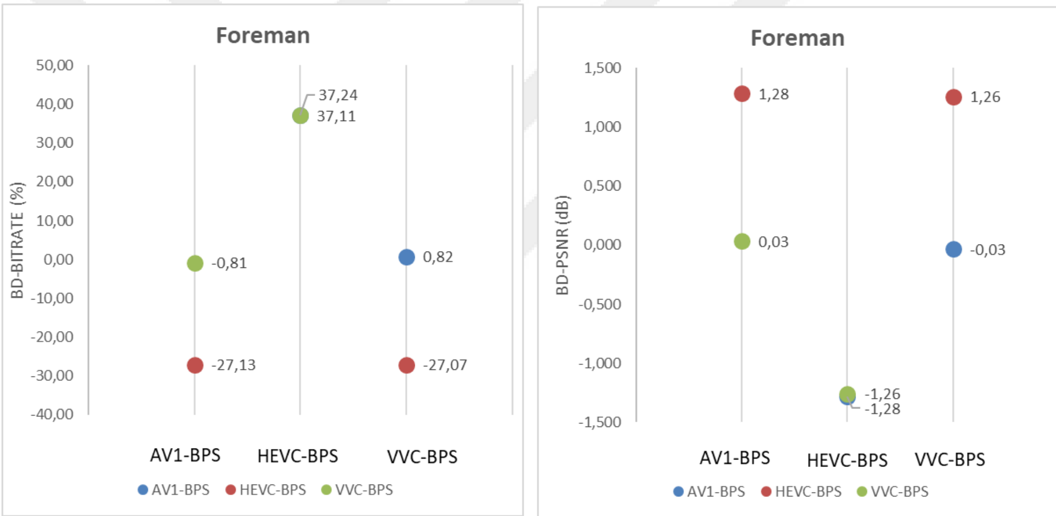

Figure 4.91. Foreman BD Rates

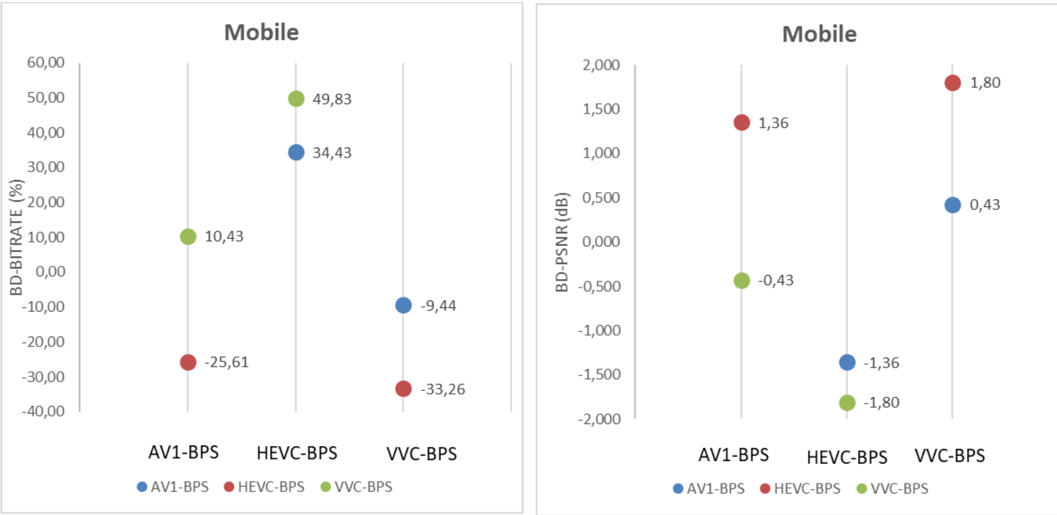

Figure 4.92. Mobile BD Rates

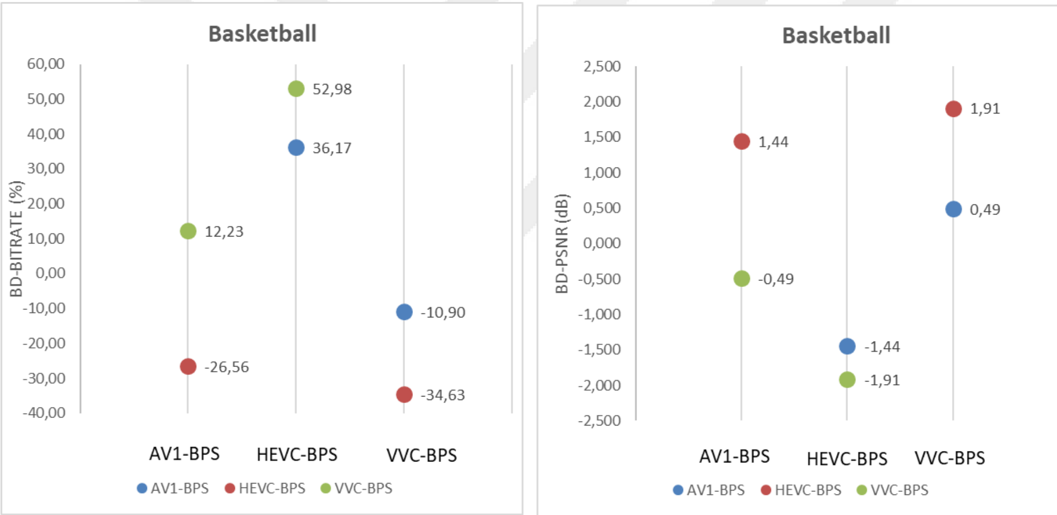

Figure 4.93. Basketball BD Rates

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, video codecs HEVC, VVC and AV1 are compared in detailed. HEVC is a well-known codec used from early 2015. On the other hand, VVC and AV1 are still being developed codecs that they believed to be the future of the codec world. These codecs are compared in terms of most essential configuration parameters of the encoding process of a video codec. Codecs are performed on 5 different characteristic CIF videos and a 4K resolution video. Results for different configurations are acquired and interpreted in terms of both PSNR and SSIM quality metrics. Finally, their comparison is held by using BD rates which provides review and in-depth analysis of rate distortion curves.

These parameters can be classified into three different parts. First part is partitioning. Coding unit size and coding unit depths are specified for HEVC and VVC, for AV1 minimum and maximum partition sizes are adjusted. For lower resolution CIF videos depth size is more important than having bigger CTU starting sizes, therefore partitioning the video into smallest CU partitions are the most important aspect for lower resolution videos. On the other hand, for 4K video, having larger CTU size is more important, because the frame resolution is large (3840,2160) partitioning every CTU into smallest possible CUs takes long time and bitrate; therefore, bigger size CUs are more efficient to encode in terms of bitrate savings. In future with 8K and beyond resolution videos become more popular, codec developers also will try to increase the biggest CTU size above 128 for future codecs.

Second parameter is I frame interval. All 3 codecs display better quality outputs as well as higher bitrate values when I frame interval gets smaller. This is expected since more I frame means more data and better quality since it is an all intra coded frame.

For HEVC and VVC bigger I frame interval results better. Quality decreases when I frame interval increases; however, the bitrate save is much better than quality loss. Overall, when BD rates and the graphs of videos are examined, it is clearly seen that higher I frame interval results with better efficiency. AV1 on the other hand, results better with lower I frame interval, this is because HEVC and VVC can efficiently code B frames and therefore with even very large I frame interval (128), they perform better by saving good number of bitrates. However, AV1 needs I frame frequently otherwise its quality decrease significantly since the reference frames that coded from Golden Frame are not coded as efficiently as HEVC or VVC.

Finally, the codecs are compared at constant bit per second configurations. For this test the 100, 200, 500, 1000 kbps codec configurations are tested for CIF videos, while for 4K video 1, 1.5, 2, 2.5 Mbps codec configurations are used. For lower complexity CIF videos AV1 performs better than VVC, and VVC performs better than HEVC. When the complexity increases VVC starts to outperform AV1, and HEVC lack behind these two codecs even more. On the other hand, for 4K video VVC is the clear winner, while the gap between HEVC and other two codecs become steeper. This indicates that HEVC starts to outdate for the brand-new resolution types such as 4K resolution and will operate even worse for future higher resolution videos such as 8K or beyond.

Future work can be implemented when VVC and AV1 have their stable versions released and started to be used by known companies or platforms. These are still very young and alive codecs. Also, the experiments can be held on a real time environment within an embedded system. By that more efficient results can be obtained for encoding times of the codecs and their usability can be decided for real time applications. Moreover, especially for VVC and AV1 there should be some improvements in order to decrease the encoding process time, especially VVC takes 20-30 times more than HEVC to encode same video with almost same configurations. The processing time is important as well as encoding efficiency especially for real time applications.

REFERENCES

[1] Vucic, D., Skorin-Kapov, L., & Sužnjević, M. (2016). The impact of bandwidth limitations and video resolution size on QoE for WebRTC-based mobile multi-party video conferencing.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper. (2019, February 18). (CISCO) Accessed on September 22, 2019 www.cisco.com: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html.

[3] Karwowski, Damian & Grajek, Tomasz & Klimaszewski, Krzysztof & Stankiewicz, Olgierd & Stankowski, Jakub & Wegner, Krzysztof. (2017). 20 Years of Progress in Video Compression – from MPEG-1 to MPEG-H HEVC. General View on the Path of Video Coding Development. 525. 3-15. 10.1007/978-3-319-47274-4_1.

[4] Le Tanou, Julien & Blestel, Mederic. (2018). Analysis of Emerging Video Codecs: Coding Tools, Compression Efficiency and Complexity.

[5] L. Wang, X. Cao, B. Niu, Q. Yu, J. Zheng and Y. He, "A Novel Derived Mode for Block Partitioning Structure in Video Coding," in IEEE Signal Processing Letters, vol. 26, no. 8, pp. 1117-1121, Aug. 2019.

[6] Ahn, YJ., Hwang, TJ., Sim, DG. et al. "Implementation of fast HEVC encoder based on SIMD and data-level parallelism," Image Video Proc (2014) 2014: 16.

[7] I. Kim, J. Min, T. Lee, W. Han and J. Park, "Block Partitioning Structure in the HEVC Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1697-1706, Dec. 2012.

[8] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[9] Y. Chen et al., "An Overview of Core Coding Tools in the AV1 Video Codec," 2018 Picture Coding Symposium (PCS), San Francisco, CA, 2018, pp. 41-45.

[10] Xu, X., & Liu, S. (2019). Recent advances in video coding beyond the HEVC standard. APSIPA Transactions on Signal and Information Processing, 8, E18.
[11] J. Lainema, F. Bossen, W. Han, J. Min and K. Ugur, "Intra Coding of the HEVC Standard" in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1792-1801, Dec. 2012.
[12] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," in IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, Jan. 1974.
[13] Soo-Chang Pei and Min-Hung Yeh, "The discrete fractional cosine and sine transforms," in IEEE Transactions on Signal Processing, vol. 49, no. 6, pp. 1198-1207, June 2001.
[14] Sarah Parker, Yue Chen, Jingning Han, Zoe Liu, Debargha Mukherjee, Hui Su, Yongzhe Wang, Jim Bankoski, Shunyao Li, "On transform coding tools under development for VP10," Proc. SPIE 9971, Applications of Digital Image Processing XXXIX, 997119 (14 October 2016).
[15] Z. Zhang, X. Zhao, X. Li, Z. Li and S. Liu, "Fast Adaptive Multiple Transform for Versatile Video Coding," 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 2019, pp. 63-72.
[16] W. Hu, G. Cheung and A. Ortega, "Intra-Prediction and Generalized Graph Fourier Transform for Image Coding," in IEEE Signal Processing Letters, vol. 22, no. 11, pp. 1913-1917, Nov. 2015.
[17] H. Sun, D. Zhou and S. Goto, "A Low-Complexity HEVC Intra Prediction Algorithm Based on Level and Mode Filtering," 2012 IEEE International Conference on Multimedia and Expo, Melbourne, VIC, 2012, pp. 1085-1090.
[18] T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh and J. Ostermann, "A Comparison of JEM and AV1 with HEVC: Coding Tools, Coding Efficiency and Complexity," 2018 Picture Coding Symposium (PCS), San Francisco, CA, 2018, pp. 36-40.
[19] Martin Alain, Christine Guillemot, Dominique Thoreau, Philippe Guillotel,Inter-prediction methods based on linear embedding for video compression,Signal Processing: Image Communication,Volume 37,2015,Pages 47-57.

[20] Y. J. Ahn, W. J. Han, D. G. Sim, "Study of decoder complexity for HEVC and AVC standards based on tool-by-tool comparison," Proc. SPIE 8499, Applications of Digital Image Processing XXXV, 84990X (15 October 2012).
[21] Pankaj Topiwala, Madhu Krishnan, Wei Dai, "Performance comparison of VVC, AV1, and HEVC on 8-bit and 10-bit content," Proc. SPIE 10752, Applications of Digital Image Processing XLI, 107520V (17 September 2018).
[22] A. Norkin et al., "HEVC Deblocking Filter," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1746-1754, Dec. 2012.
[23] V. Sze and M. Budagavi, "High Throughput CABAC Entropy Coding in HEVC," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1778-1791, Dec. 2012.
[24] W. G. Shadeed, D. I. Abu-Al-Nadi and M. J. Mismar, "Road traffic sign detection in color images," 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003, Sharjah, 2003, pp. 890-893 Vol.2.
[25] YUV Video Sequences. Accessed on October 03, 2018 Video Trace Library: http://trace.eas.asu.edu/yuv/index.html.
[26] Okamoto, J., Hayashi, T., Takahashi, A. and Kurita, T. (2006), Proposal for an objective video quality assessment method that takes temporal and spatial information into consideration. Electron. Comm. Jpn. Pt. I, 89: 97-108.
[27] T. K. Tan et al., "Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 76-90, Jan. 2016.
[28] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," 2010 20th International Conference on Pattern Recognition, Istanbul, 2010, pp. 2366-2369.
[29] Tianyi Wang, A. Pervez and Hua Zou, "VQM-based QoS/QoE mapping for streaming video," 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), Beijing, 2010, pp. 807-812.

[30] Generic Coding of Moving Pictures and Associated Audio Information— Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1, 1994.
[31] P. Hanhart and T. Ebrahimi, "Calculation of average coding efficiency based on subjective quality scores" EPFL, Switzerland.
[32] Y. Zhang, H. Wang and Z. Li, "Fast Coding Unit Depth Decision Algorithm for Interframe Coding in HEVC," 2013 Data Compression Conference, Snowbird, UT, 2013, pp. 53-62.
[33] J. Leng, L. Sun, T. Ikenaga and S. Sakaida, "Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC," 2011 International Conference on Multimedia and Signal Processing, Guilin, Guangxi, 2011, pp. 56-59.
[34] F. Mu, L. Song, X. Yang and Z. Luo, "Fast coding unit depth decision for HEVC," 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, 2014, pp. 1-6.
[35] X. Shen, L. Yu and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," 2012 Picture Coding Symposium, Krakow, 2012, pp. 453-456.
[36] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesana, A. Piva and M. Barni, "Detection of video double encoding with GOP size estimation," 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Tenerife, 2012, pp. 151-156.
[37] Xiaodong Gu and Hongjiang Zhang, "Implementing dynamic GOP in video encoding," 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698), Baltimore, MD, USA, 2003, pp. I-349.
[38] L. Czuni, G. Csaszar and A. Licsar, "Estimating the Optimal Quantization Parameter in H.264," 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, 2006, pp. 330-333.
[39] T. Zhao, Z. Wang and C. W. Chen, "Adaptive Quantization Parameter Cascading in HEVC Hierarchical Coding," in IEEE Transactions on Image Processing, vol. 25, no. 7, pp. 2997-3009, July 2016.

APPENDICES

A. HEVC Codec Configurations

Table 0.1. *HEVC Random Access Configurations*

Encoder Setting	Default Value	Description
BitstreamFile		Input File
ReconFile		Output File
MaxCUWidth	64	Maximum coding unit width in pixel
MaxCUHeight	64	Maximum coding unit height in pixel
MaxPartitionDepth	4	Maximum coding unit depth
QuadtreeTULog2MaxSize	5	Log2 of maximum transform size for quadtree-based TU coding (2...6)
QuadtreeTULog2MinSize	2	Log2 of minimum transform size for quadtree-based TU coding (2...6)
QuadtreeTUMaxDepthInter	3	
QuadtreeTUMaxDepthIntra	3	
IntraPeriod	32	Period of I-Frame
DecodingRefreshType	1	Random Accesss 0:none, 1:CRA, 2:IDR, 3:Recovery Point SEI
GOPSize	16	GOP Size
ReWriteParamSetsFlag	1	Write parameter sets with every IRAP

Encoder Setting	Default Value	Description
IntraQPOffset	-3	
LambdaFromQpEnable	1	
FastSearch	1	0:Full search 1:TZ search
SearchRange	384	0: Search range is a Full frame
ASR	1	Adaptive motion search range
MinSearchWindow	96	Minimum motion search window size for the adaptive window ME
BipredSearchRange	4	Search range for bi-prediction refinement
HadamardME	1	Use of hadamard measure for fractional ME
FEN	1	Fast encoder decision
FDM	1	Fast Decision for Merge RD cost
QP	32	Quantization parameter(0-51)
MaxDeltaQP	0	CU-based multi-QP optimization
MaxCuDQPDepth	0	Max depth of a minimum CuDQP for sub-LCU-level delta QP
DeltaQpRD	0	Slice-based multi-QP optimization
RDOQ	1	
RDOQTS	1	RDOQ for transform skip
SliceChromaQPOffsetPeriodicity	0	
SliceCbQpOffsetIntraOrPeriodic	0	

Encoder Setting	**Default Value**	**Description**
SliceCrQpOffsetIntraOrPeriodic	0	
LoopFilterOffsetInPPS	1	
LoopFilterDisable	0	Disable deblocking filter (0=Filter, 1=No Filter)
LoopFilterBetaOffset_div2	0	Base_param: -6 ~ 6
LoopFilterTcOffset_div2	0	Base_param: -6 ~ 6
DeblockingFilterMetric	0	Blockiness metric
InternalBitDepth	8	codec operating bit-depth
SAO	1	Sample adaptive offset (0: OFF, 1: ON)
AMP	1	Asymmetric motion partitions (0: OFF, 1: ON)
TransformSkip	1	Transform skipping (0: OFF, 1: ON)
TransformSkipFast	1	Fast Transform skipping (0: OFF, 1: ON)
SAOLcuBoundary	0	SAOLcuBoundary using non-deblocked pixels (0: OFF, 1: ON)
SliceMode	0	
SliceArgument	1500	
LFCrossSliceBoundaryFlag	1	
PCMEnabledFlag	0	0: No PCM mode
PCMLog2MaxSize	5	Log2 of maximum PCM block size
PCMLog2MinSize	3	Log2 of minimum PCM block size

Encoder Setting	Default Value	Description
PCMInputBitDepthFlag	1	0: PCM bit-depth is internal bit-depth. 1: PCM bit-depth is input bit-depth
PCMFilterDisableFlag	0	0: Enable loop filtering on I_PCM samples. 1: Disable loop filtering on I_PCM samples
TileUniformSpacing	0	
NumTileColumnsMinus1	0	
TileColumnWidthArray	2 3	
NumTileRowsMinus1	0	
TileRowHeightArray	2	
LFCrossTileBoundaryFlag	1	
WaveFrontSynchro	0	
ScalingList	0	ScalingList 0 : off, 1 : default, 2 : file read
ScalingListFile	*.txt	Scaling List file name
TransquantBypassEnableFlag	0	Value of PPS flag
CUTransquantBypassFlagForce	0	Force transquant bypass mode, when transquant_bypass_enable_flag is enabled
RateControl	0	Enable rate control
TargetBitrate	1000000	Target bitrate, in bps
KeepHierarchicalBit	2	0: equal bit allocation; 1: fixed ratio bit allocation; 2: adaptive ratio bit allocation

Encoder Setting	Default Value	Description
LCULevelRateControl	1	1: LCU level RC; 0: picture level RC
RCLCUSeparateModel	1	Use LCU level separate R-lambda model
InitialQP	0	Initial QP
RCForceIntraQP	0	Force intra QP to be equal to initial QP

B. VVC Codec Configurations

Table 0.2. *VVC Random Access Configurations*

Encoder Setting	Default Value	Description
BitstreamFile		Input File
ReconFile		Output File
MaxCUWidth	64	Maximum coding unit width in pixel
MaxCUHeight	64	Maximum coding unit height in pixel
MaxPartitionDepth	4	Maximum coding unit depth
IntraPeriod	32	Period of I-Frame
DecodingRefreshType	1	Random Accesss 0:none, 1:CRA, 2:IDR, 3:Recovery Point SEI
GOPSize	16	GOP Size
IntraQPOffset	-3	
LambdaFromQpEnable	1	
FastSearch	0	0:Full search 1:TZ search
SearchRange	384	(0: Search range is a Full frame)
ASR	1	Adaptive motion search range
MinSearchWindow	96	Minimum motion search window size for the adaptive window ME
BipredSearchRange	4	Search range for bi-prediction refinement
HadamardME	1	Use of hadamard measure for fractional ME

Encoder Setting	Default Value	Description
FEN	1	Fast encoder decision
FDM	1	Fast Decision for Merge RD cost
QP	32	Quantization parameter(0-51)
MaxDeltaQP	0	CU-based multi-QP optimization
MaxCuDQPSubdiv	0	Maximum subdiv for CU luma Qp adjustment
DeltaQpRD	0	Slice-based multi-QP optimization
RDOQ	1	
RDOQTS	1	RDOQ for transform skip
LoopFilterOffsetInPPS	1	
LoopFilterDisable	0	Disable deblocking filter (0=Filter, 1=No Filter)
LoopFilterBetaOffset_div2	0	
LoopFilterTcOffset_div2	0	
DeblockingFilterMetric	0	blockiness metric
InternalBitDepth	10	codec operating bit-depth
SAO	1	Sample adaptive offset (0: OFF, 1: ON)
TransformSkip	1	Transform skipping (0: OFF, 1: ON)
TransformSkipFast	1	Fast Transform skipping (0: OFF, 1: ON)
TransformSkipLog2MaxSize	5	

Encoder Setting	Default Value	Description
SAOLcuBoundary	0	
SliceMode	0	
SliceArgument	1500	
LFCrossSliceBoundaryFlag	1	
PCMEnabledFlag	0	0: No PCM mode
PCMLog2MaxSize	5	Log2 of maximum PCM block size
PCMLog2MinSize	3	Log2 of minimum PCM block size
PCMInputBitDepthFlag	1	0: PCM bit-depth is internal bit-depth. 1: PCM bit-depth is input bit-depth
PCMFilterDisableFlag	0	0: Enable loop filtering on I_PCM samples. 1: Disable loop filtering on I_PCM samples
TransquantBypassEnable	0	Value of PPS flag
CUTransquantBypassFlagForce	0	Force transquant bypass mode
RateControl	0	Enable rate control
TargetBitrate	1000000	Target bitrate, in bps
KeepHierarchicalBit	2	0: equal bit allocation; 1: fixed ratio bit allocation; 2: adaptive ratio bit allocation
LCULevelRateControl	1	1: LCU level RC; 0: picture level RC
RCLCUSeparateModel	1	Use LCU level separate R-lambda model

Encoder Setting	**Default Value**	**Description**
InitialQP	0	Initial QP
RCForceIntraQP	0	Force intra QP to be equal to initial QP
LoopFilterTcOffset_div2	0	
SEIDecodedPictureHash	0	
CbQpOffset	1	
CrQpOffset	1	
ReWriteParamSets	1	
CTUSize	128	
LCTUFast	1	
DualITree	1	
MinQTLumaISlice	8	
MinQTChromaISlice	4	
MinQTNonISlice	8	
MaxMTTHierarchyDepth	3	
MaxMTTHierarchyDepthISliceL	3	
MaxMTTHierarchyDepthISliceC	3	
MTS	1	
MTSIntraMaxCand	3	
MTSInterMaxCand	4	
SBT	1	
LFNST	1	
ISP	1	
MMVD	1	

Encoder Setting	Default Value	Description
Affine	1	
SubPuMvp	1	
MaxNumMergeCand	6	
LMChroma	1	
DepQuant	1	
IMV	1	
ALF	1	
GBi	1	
GBiFast	1	
BIO	1	
MHIntra	1	
Triangle	1	
IBC	0	
AllowDisFracMMVD	1	
AffineAmvr	1	
LumaReshapeEnable	1	
MIP	1	
DMVR	1	
SMVD	1	
PBIntraFast	1	
ISPFast	1	
FastMrg	1	
AMaxBT	1	
FastMIP	0	

Encoder Setting	Default Value	Description
FastLFNST	0	
AffineAmvrEncOpt	1	
MmvdDisNum	6	

C. AV1 Codec Configurations

Table 0.3. *AV1 Configuration Parameters*

Encoder Setting	Default Value	Description
--help		Show usage options and exit
--cfg=<arg>		Config file to use
--debug		Debug mode
--output=<arg>		Output filename
--codec=<arg>		Codec to use
--passes=<arg>		Number of passes (1/2)
--pass=<arg>		Pass to execute (1/2)
--fpf=<arg>		First pass statistics file name
--limit=<arg>		Stop encoding after n input frames
--skip=<arg>		Skip the first n input frames
--good		Use Good Quality Deadline
--rt		Use Realtime Quality Deadline
--quiet		Do not print encode progress
--verbose		Show encoder parameters
--psnr		Show PSNR in status line
--webm		Output WebM (default when WebM IO is enabled)
--ivf		Output IVF
--obu		Output OBU
--q-hist=<arg>		Show quantizer histogram (n-buckets)

Encoder Setting	Default Value	Description
--rate-hist=<arg>		Show rate histogram (n-buckets)
--disable-warnings		Disable warnings about potentially incorrect encode settings
--disable-warning-prompt		Display warnings, but do not prompt user to continue
--test-decode=<arg>		Test encode/decode mismatch
--yv12		Input file is YV12
--i420		Input file is I420
--i422		Input file is I422
--i444		Input file is I444
--usage=<arg>		Usage profile number to use
--threads=<arg>		Max number of threads to use
--profile=<arg>		Bitstream profile number to use
--width=<arg>		Frame width
--height=<arg>		Frame height
--forced_max_frame_width		Maximum frame width value to force
--forced_max_frame_height		Maximum frame height value to force
--stereo-mode=<arg>		Stereo 3D video format
--timebase=<arg>		Output timestamp precision (fractional seconds)
--fps=<arg>		Stream frame rate (rate/scale)
--global-error-resilient =< -b <arg>, --bit-depth=<arg>		Enable global error resiliency features

Encoder Setting	**Default Value**	**Description**
--bit-depth=<arg>		Bit depth for codec (8 for version <=1, 10 or 12 for version 2)
--lag-in-frames=<arg>		Max number of frames to lag
--large-scale-tile=<arg>		Large scale tile coding (0: off (default), 1: on (ivf output only))
--monochrome		Monochrome video (no chroma planes)
--full-still-picture-hdr		Use full header for still picture
--drop-frame=<arg>		Temporal resampling threshold (buf %)
--resize-mode=<arg>		Frame resize mode
--resize-denominator=<arg>		Frame resize denominator
--resize-kf-denominator=<arg>		Frame resize keyframe denominator
--superres-mode=<arg>		Frame super-resolution mode
--superres-denominator=<arg>		Frame super-resolution denominator
--superres-kf-denominator=<arg>		Frame super-resolution keyframe denominator
--superres-qthresh=<arg>		Frame super-resolution qindex threshold
--superres-kf-qthresh=<arg>		Frame super-resolution keyframe qindex threshold
--end-usage=<arg>		Rate control mode
--target-bitrate=<arg>		Bitrate (kbps)
--min-q=<arg>		Minimum (best) quantizer

Encoder Setting	Default Value	Description
--max-q=<arg>		Maximum (worst) quantizer
--undershoot-pct=<arg>		Datarate undershoot (min) target (%)
--overshoot-pct=<arg>		Datarate overshoot (max) target (%)
--buf-sz=<arg>		Client buffer size (ms)
--buf-initial-sz=<arg>		Client initial buffer size (ms)
--buf-optimal-sz=<arg>		Client optimal buffer size (ms)
--bias-pct=<arg>		CBR/VBR bias (0=CBR, 100=VBR)
--minsection-pct=<arg>		GOP min bitrate (% of target)
--maxsection-pct=<arg>		GOP max bitrate (% of target)
--enable-fwd-kf=<arg>		Enable forward reference keyframes
--kf-min-dist=<arg>		Minimum keyframe interval (frames)
--kf-max-dist=<arg>		Maximum keyframe interval (frames)
--disable-kf		Disable keyframe placement
--cpu-used=<arg>		CPU Used (0..8)
--auto-alt-ref=<arg>		Enable automatic alt reference frames
--sharpness=<arg>		Loop filter sharpness (0..7)
--static-thresh=<arg>		Motion detection threshold
--row-mt=<arg>		Enable row based multi-threading (0: off, 1: on (default))

Encoder Setting	Default Value	Description
--tile-columns=<arg>		Number of tile columns to use, log2
--tile-rows=<arg>		Number of tile rows to use, log2
--enable-tpl-model=<arg>		RDO based on frame temporal dependency
--arnr-maxframes=<arg>		AltRef max frames (0..15)
--arnr-strength=<arg>		AltRef filter strength (0..6)
--tune=<arg>		Distortion metric tuned with psnr, ssim
--cq-level=<arg>		Constant/Constrained Quality level
--max-intra-rate=<arg>		Max I-frame bitrate (pct)
--max-inter-rate=<arg>		Max P-frame bitrate (pct)
--gf-cbr-boost=<arg>		Boost for Golden Frame in CBR mode (pct)
--lossless=<arg>		Lossless mode (0: false (default), 1: true)
--min-partition-size=<arg>		Set min partition size (4:4x4, 8:8x8, 16:16x16, 32:32x32, 64:64x64, 128:128x128)
--max-partition-size=<arg>		Set max partition size (4:4x4, 8:8x8, 16:16x16, 32:32x32, 64:64x64, 128:128x128)