Scalable Video codec for High Efficiency Video Codec (SVT-HEVC) Software Encoder for 5G Edge Cloud.

Submitted By

Darshali Patel

17MCEN11

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

Scalable Video codec for High Efficiency Video Codec (SVT-HEVC) Software Encoder for 5G Edge Cloud.

Major Project

Submitted in partial fulfillment of the requirements for the

degree of

Master of Technology in Computer Science and Engineering (Networking Technology)

Submitted By

Darshali Patel

(17MCEN11)

Guided By

Prof. Vipul Chudasama

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

DEPARTMENT OF COMPUTER ENGINEERING

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

Certificate

This is to certify that the major project entitled" **Scalable Video Technology for High Efficiency Video Codec (SVT-HEVC) Software Encoder for 5G Edge Cloud."** submitted by **Darshali Patel (Roll No: 17MCEN11)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Specialization in Networking Technology) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Vipul Chudasama
Guide & Assistant Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.Gaurang Raval
Associate Professor,
Coordinator M.Tech - CSE
(Networking Technology)
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
IT Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Darshali Patel**, Roll. No. **17MCEN11**, give undertaking that the Major Project entitled" **Scalable Video codec for High Efficiency Video Codec (SVT-HEVC) Software Encoder for 5G Edge Cloud**." submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Networking Technology)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

————————

Signature of Student

Date:

Place:

Endorsed by

Prof. Vipul Chudasama

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Vipul Chudasama,** Assistant Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement through- out this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar,** Hon'ble Head of Information Technology Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Darshali Patel

17MCEN11

Abstract

High Efficiency Video Coding is at present being set up as the most up to date video coding Standard of the ITU-T Video coding expert Group and the ISO/IEC Moving picture expert Group. The principle objective of the HEVC standardization effort is enable to significantly increase Compression performance relative to 50 Percentage of bitrate for equal video Quality.

But the problem with now what we have is Ever associated with WI-FI organize just to discover its slower than your 4G. With the end goal to set a side extra cash, numerous spots decide on less expensive Internet Design with their ISPs which results in moderate transfer speeds. This powers numerous live stream tasks at areas like schools and holy places to stream at lower bitrates and goals, bringing about a low-quality feed not exp tremely speaking to watchers.There for this report has problem solving implementation with the use of many techniques of HEVC standard.I include the results of the SVT-HEVC Application. Which could be the increase the performance of the video Coding Compression.

Abbreviations

SVT-HEVC	Scalable Video Technology for High Efficiency Video Coding
CTU	Control Tree Unit
PB	Pridiction Blocks
CB	Coding Blocks
SAO	Sample Adaptive Offsets
BAC	Binary Arithmetic Coding
TB	Transform Block

Contents

List of Figures

Chapter 1 :Introduction

This Report describes why we are implementing this scalable Video Technology for High efficiency Video Coding and what are the techniques using in this particular technology. H.264 is also known as Advanced Video Coding (AVC). It is the previous video coding standard which is block oriented motion compensation based video compression standard. It is mainly use for video Recording, Compression and distribution video efficiency.H.264 is support 3840*2160 Resolution for all video.

High Efficiency Video Coding is standard latest Codec for boost up video streaming and it is video Compression standard for increase efficiency in any type of Video. HEVC support 8192*4320 with 8K ultra HD resolution.

In this Project our goal is to increase scalability of the video with the help of this HEVC encoder and also planning to get more reliable result of encoding video which is not more reliable and scalable now a days. For example if people are seeing live match

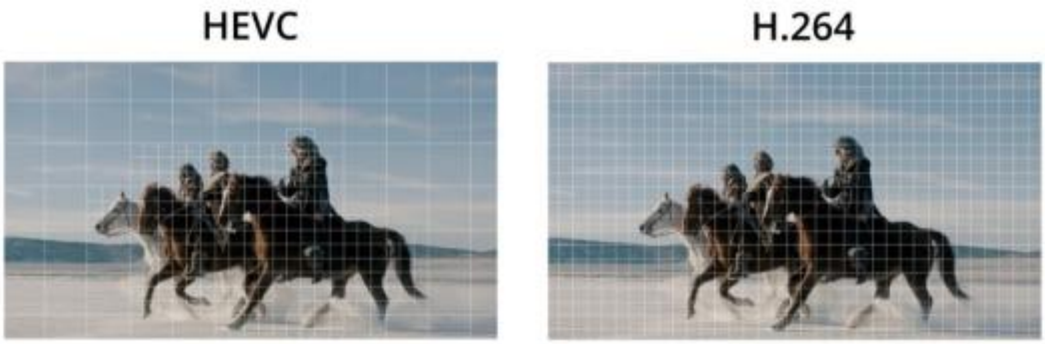


Figure 1.1: High Efficiency Video Coding vs H.264[8]

Figure 1.2: [a] [b] [c]

on the TV so camera man is captured that match video and store into device and then passing to our home so the process is time consuming and result what people get is not clear.

There for the HEVC is decrease the time between broadcasting and increase the scalT ability of video with use of encryption and decryption. Also we compress the video with the use of algorithms and encoding Techniques.

Basically Video Compression is more accurate in HEVC which is works by eliminating redundancy and irrelevancy. Redundant information is as of now present elsewhere in the record and can be acquired from that point as opposed to putting away it again. Since the progressions between two back to back pictures are regularly small, the majority of the data in the second picture is redundant. Evacuating repetition results in lossless compression.

In contrast to redundancy, superfluity is a relative idea. While evacuating insignificant data, we begin by expelling subtleties that are unnoticeable to the HVS. Be that as it may, so as to accomplish an objective bitrate/record estimate, we may need to begin evacuating data of more prominent significance, which perceptibly corrupts the video quality. Regardless, we try to remove data so that the nature of the subsequent video is maximized.

Evacuating redundancy is clearly superior to expelling irrelevancy, since the video quality is unaffected. Be that as it may, there is just a constrained measure of redundancy and keeping in mind that it is somewhat simple to discover a few, it turns out to be progressively troublesome the more we endeavor to crush out. Along these lines, each time a video is encoded, there is a tradeoff between three parameters: quality, bitrate and encoding runtime.

1.1 SVT-HEVC

Scalable Video Technology for HEVC Encoder (SVT-HEVC) Encoder utilizes a packed 10-bit arrange enabling the product to accomplish a higher speed and channel thickness levels. The transformation between the 10-bit yuv420p10le and the packed 10-bit arrange is a lossless activity.

Video Compression: Video Compression is the encoding method which can be use the less space than real video sample and it can be easily transfer to the network.with the help of this video compression it can decrease the size of video file.

Techniques of high efficiency video coding:

- Coding Tree Structure
- Inter Prediction
- Intra Prediction
- Motion Vector coding

1.1.1 Coding Tree Structure

CTU: Coding Tree Unit is therefore a logical unit. It usually consists of three blocks, namely luma (Y) and two Chroma samples (Cb and Cr), and associated syntax elements. Each block is called CTB (Coding Tree Block).

Each CTB still has unclear size from CTU 64*64, 32*32, or 16*16. Dependent upon a bit of video diagram, in any case, CTB may be excessively gigantic, making it impossible to pick whether we should perform between picture desire or intra picture conjecture. Thusly, each CTB can be contrastingly part into various CBs (Coding Blocks) and each CB transforms into the essential initiative reason for conjecture type. For example, some CTBs are part to 16*16 CBs while others are part to 88 CBs. HEVC supports CB gauge the separation from unclear size from CTB to as meager as 8*8.

CB is sufficient for Control Blocks type , in any case it could in any case be too much impressive, influencing it difficult to store advancement vectors (to bury expectation) or intra prediction mode. For instance, a size of blocks may move amidst 8*8 CB we need to utilize specific MVs relying on the section in CB

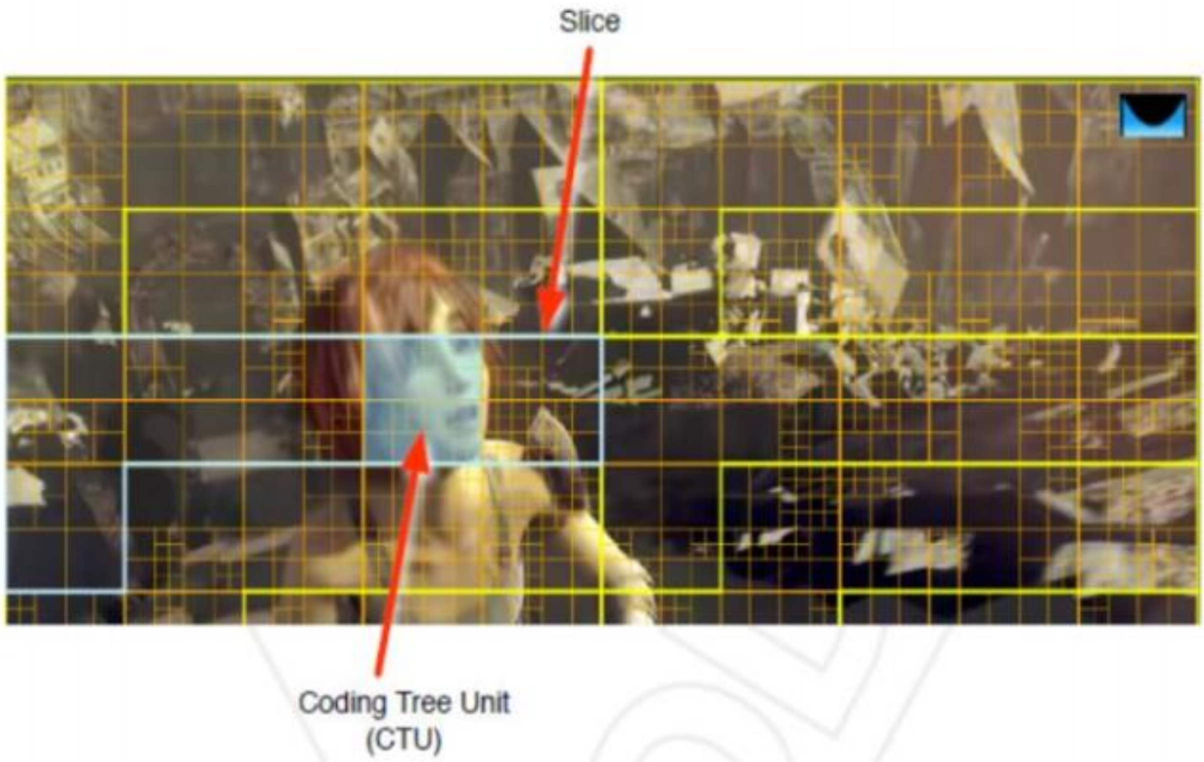


Figure 1.3: CTU- Control tree unit Structure[6]

A coding tree square of luma tests, two comparing coding tree squares of chroma tests of an image that has three example clusters, or a coding tree square of tests of Inter picture or an image that is coded utilizing three separate shading planes and sentence structure structures used to code the examples.

The width and height of CTU are signaled in a sequence parameter set, meaning that all the CTUs in a video sequence have the same size: 64*64, 32*32, or 16*16.CTB:Coding Tree Blocks- A CTB may contain only one CU or may be split to form multiple CUs, and each CU has an associated partitioning into prediction units (PUs) and a tree of transform units (TUs).

CB: Coding Blocks The quadtree language structure of the CTU determines the size and places of its luma and chroma CBs. The foundation of the quadtree is related with the CTU.Thus, the extent of the luma CTB is the biggest bolstered estimate for a luma CB. The part of a CTU into luma and chroma CBs is flagged mutually. One luma CB and commonly two chroma CBs, together with related linguistic structure, frame a coding unit (CU). A CTB may contain just a single CU or might be part to frame various CUs, and every CU has a related parceling into prediction units (PUs) and a tree of change units (TUs).

PB: Prediction Block The prediction Blocks isolating Structure has its root at CU levels.Dependent upon the crucial gauge type decision, the luma and chroma CBs would then have the capacity to be moreover part in size and foreseen from luma and chroma desire squares (PBs). HEVC supports variable PB sizes from 64*64 down to 4*4 preced dents.Coefficient sifting is performed in 4*4 subblocks for all TB sizes.

1.1.2 Inter Prediction

Inter-picture prediction in HEVC can be viewed as a relentless enhancement and specI ulation of all parts known from past video coding benchmarks, e.g. H.264/AVC. The movement vector expectation was improved with cutting edge movement vector forecast dependent on movement vector rivalry. An Inter-picture prediction combining procedure altogether improved the square astute movement information motioning by gathering all. movement information from as of now decoded squares. With regards to addition of partial reference picture tests, high exactness insertion channel portions with broadened bolster,i.e. 7/8-tap channel parts for luma and 4-tap channel bits for chroma, enhance the sifting particularly in the high recurrence extend. At long last, the weighted expect tation flagging was improved by either applying unequivocally flagged weights for each movement repaid forecast or simply averaging two movement remunerated expectations.

This part gives the information including video file and quality of these parts of HEVC standard and clarifies their coding productivity and intricacy qualities. Each PU is predicted from image data in one or two reference pictures (before or after the current picture in display order), using motion compensated prediction. Motion vectors have up to quarter sample resolution luma prediction)

In picture or video compression it is completely sensible to send a square based picture isolating framework with the ultimate objective to apply assorted prediction models to different zones of an image. This is in light of the fact that a lone model can when all is said in done not be depended upon to get the adaptable characteristics of a whole picture or video.

HEVC uses a quad-tree structure to delineate the dividing a territory into sub squares. To the extent bit rate, this is an insignificant exertion structure while meanwhile, it considers partitioning into a broad assortment of differently assessed sub-squares. While this straightforwardness is great position for example for encoder structure, it furthers more bears the shortcoming of over-dividing the image, potentially provoking redundant signalling and insufficient edges.

1.1.3 Intra Prediction

Intrapicture prediction works as per the TB estimate, and recently decoded limit tests from spatially neighboring TBs are utilized to frame the prediction signal.Directional prediction with 33 distinctive directional introductions is characterized for (square) TB sizes from 44 up to 32*32. For chroma, the flat, vertical, planar, and DC prediction modes can be expressly flagged, or the chroma prediction mode can be demonstrated to be equivalent to the luma prediction mode.

The intra prediction framework of HEVC consists of three steps: reference sample array construction, sample prediction, and post-processing. All the three steps have been designed to achieve high coding efficiency while minimizing the computational required ments in both the encoder and decoder. The set of defined prediction modes consists of methods modeling various types of content typically present in video and still imo ages. The HEVC angular prediction provides high-fidelity predictors for objects with directional structures, and the additional planar and DC prediction modes can effectively model smooth image areas.

1.1.4 Motion Vector Coding

MV coding in HEVC, particularly the derivation strategies of movement candidates for the inter mode, skip mode, and merge mode, in this report, a general review of MV prediction and MV coding methods is first displayed. In addition, this paper likewise depicts our few proposed MV prediction and MV coding methods, which were received into the standard. The proposed strategies incorporate a priority-based derivation calculation for spatial movement candidates, a priority based derivation calculation for worldly movement candidates, an encompassing based candidate rundown, and parallel derivation of the candidate list. There are two strategies in Motion vector coding one is combine mode and another Is Advanced Motion Vector Prediction.Motion vector can be determined in Merge/skip mode. Defines a pursuit window focus purpose of information prediction Unit in the motion Estimation process utilizing srounding accessible Motion Vectors.

Chapter 2 : Video Coding Standard

The coming of digital TV, the presence of digital versatile disks (DVDs), and the exchange of video documents over the Internet exhibit the significance of digital video. The utilization of picture and video applications, for example, the World Wide Web and video conferencing has expanded drastically as of late. At the point when correspondence transmission capacity or capacity is restricted, information is packed. Particularly when a remote system is used, low bit rate compression calculations are required as a result of the constrained transmission capacity. The point of video source coding is bit-rate decrease proposed for transmission and capacity by investigating both measurable and abstract redundancies, and to encode a "base set" of data by applying entropy coding methods. This typically results in a compression of the coded video information contrasted with the first source information.

2.1 Past Video Coding Standard :

In the media communications world, universal benchmarks are most significant in light of the fact that correspondence on a worldwide scale is incomprehensible without interoperability of hardware from various producers and vendors. Several diverse video coding have been created in recent decades. These norms imitate the mechanical advancement in video coding and the adjustment of video coding guidelines to various applications and systems.

2.2 Advanced Vector Coding (AVC)/MPEG4:

H.264/AVC standard has been created and institutionalized in the coordinated effort of both the ISO/IEC MPEG and ITU-T VCEG associations. It is the most generally utilized video coding standard for gushing videos, mobile/handheld applications, HDTV broadcasting etc. Compared to the generally contemplated MPEG-4 encoded video, the H.264 video coding models incorporate novel components, for instance various leveled B outline prediction structures and highly effective adaptable coding that have critical ramifications for system transport [9]. A standout amongst the most clear contrasts from more seasoned measures is its expanded adaptability for bury coding. H.264 can accomplish a bit-rate decrease of up to 50 percent conversely with MPEG-4 advanced straightforward profile [8]. This improvement originates from the prediction stage and the motion estimation (ME) at quarter-pixels with variable square sizes (VBSs) and numerous reference outlines.

H.264 standard characterizes a lot of profiles to set purposes of conformance for various classes of uses and administrations. A subset of the whole piece stream grammar or in different terms a subset of the coding apparatuses is characterized as profile [9].The H.264 standard incorporates seven profiles, focusing on explicit classes of utilizations. It likewise has four extra all-Intra profiles, which are generally utilized for expert applications. In addition, the H.264 Scalable Video Coding (SVC) standard incorporates three profiles which empower adaptability by encoding a video stream into various layers each with various quality data. SVC Empowers distinctive nature of experience for various clients based on the quantity of sub streams they get [7]. The encoded scalable video can be decoded in various characteristics. Fifteen dimensions are indicated in H.264/AVC for each profile. Each dimension determines lower limits for the decoder capacities or upper limits for the bit stream.

Rather than past video coding guidelines, the Intra prediction in H.264/AVC is constantly directed in the spatial space, by indicating neighboring examples of recently decoded obstructs that are to one side and additionally over the square to be anticipated. Motion estimation (ME) is the most tedious segment in the H.264 video coding standard. Because of its high computational unpredictability, motion estimation expends 70% to 90% of the complete encoding time. H.264 utilizes a variable square size for entomb mode prediction, which has 7 different MB coding modes so the fleeting and spatial subtleties in a MB are best introduced. The H.264/AVC bolsters motion compensated prediction through quarter pel exactness of motion vectors. In H.264 video coding standard, the different reference outline motion estimation is bolstered to get a superior prediction result. In this manner, the motion vectors inside a solitary casing may point to various reference outlines. Every video outline (picture) in H.264 is either intra coded (I), forward prescient coded (P) with motion remunerated prediction from the previous I or P edge, or bi-directionally prescient coded (B) as per its situation in a group of pictures (GOP) structure.

2.3 High Efficiency Video Coding /HEVC:

Work has been done on upgrading the coding standard, either by lessening the encoding time, multifaceted nature, improving the quality, or improving the heartiness of the benchmarks utilizing calculations for mistake disguise and blunder redress [20]. HEVC is the name of the present joint institutionalization venture of ISO/IEC MPEG and ITU-T VCEG, created in a cooperation known as the Joint Collaborative Team on Video Coding (JCT-VC). The standard has been concluded in 2013. It gives higher coding execution than past state of-the-craftsmanship video coding standard H.264/AVC. However, the close half piece rate decrease contrasted with H.264/AVC primarily originates from the reception of a few advanced coding procedures yet to the detriment of expanded coding multifaceted nature just as equipment structure trouble.

In HEVC, an image is apportioned into coding tree squares (CTBs). Like H.264/MPEG-4 AVC, HEVC underpins quarter test exactness motion vectors. HEVC likewise underpins numerous reference pictures, and the ideas of I, P, and B cuts are essentially unaltered from H.264/MPEG-4 AVC. Weighted prediction is likewise bolstered along these lines. Consolidation mode motion vector rivalry approach is utilized in HEVC to infer motion vector prediction. Here, motion vector rivalry is that the best motion vector with least rate bending cost is chosen from the motion vector applicants. Prediction is made utilizing the advanced motion vector prediction (AMVP) calculation. All cut information linguistic structure components are entropy coded with CABAC, which is practically equivalent to the CABAC coding in H.264/MPEG-4 AVC.Besides a deblocking channel, the HEVC configuration incorporates a Sample Adaptive Offset (SAO) task inside the motion compensation loop.

For greatest measure of interoperability between gadgets the quantity of profiles ought to be less. The HEVC determines three profiles focusing on various application prerequisites, e.g., Main, Main 10, Main Still Picture profiles. In Main profile (MP), the coding of 8-bit-persample video in 4:2:0 Chroma positions is bolstered. The Main 10 profile backings coding of 10-bit-per-test video. Inter picture prediction isn't bolstered in the Main Still Picture profile, since it requires just a solitary one coded picture in whole bit stream.

Chapter 3 : Literature Survey

3.1 Techniques

3.1.1 High level Parallelism in HEVC

HEVC provides several low- and high-level mechanisms for dependency removal whereby only the latter are of interest for this work [1]. Namely these are slices, tiles and WPP, which all subdivide the video frames in certain ways. In general, every picture is subdiw vided into CTUs which have a maximum size of 64*64 luma pixels. They can be recurv sively split into square-shaped Coding Units (CUs), which in turn contain one or more PUs and Transform Units. It is important to note that all the high-level parallelization schemes operate at the CTU level [2].

3.1.2 Loop Filtering

In HEVC, in-loop filtering comprises of deblocking channel and SAO. It is connected to the reproduced pixels just before thinking of them into the decoded picture cradle inside the bury forecast circle. Like that in H.264/AVC, deblocking channel in HEVC intends to lessen the perceivability of blocking relics caused by square based coding structure and is connected just to tests situated at square limits [3]. Though, SAO is a recently embraced apparatus in HEVC, which is planned to enhance precision of recreation of the first flag. SAO is performed adaptively for all examples by restrictively increasing the value of each example dependent on qualities in look-into tables characterized by HEVC encoder.

3.1.3 Deblocking Filter

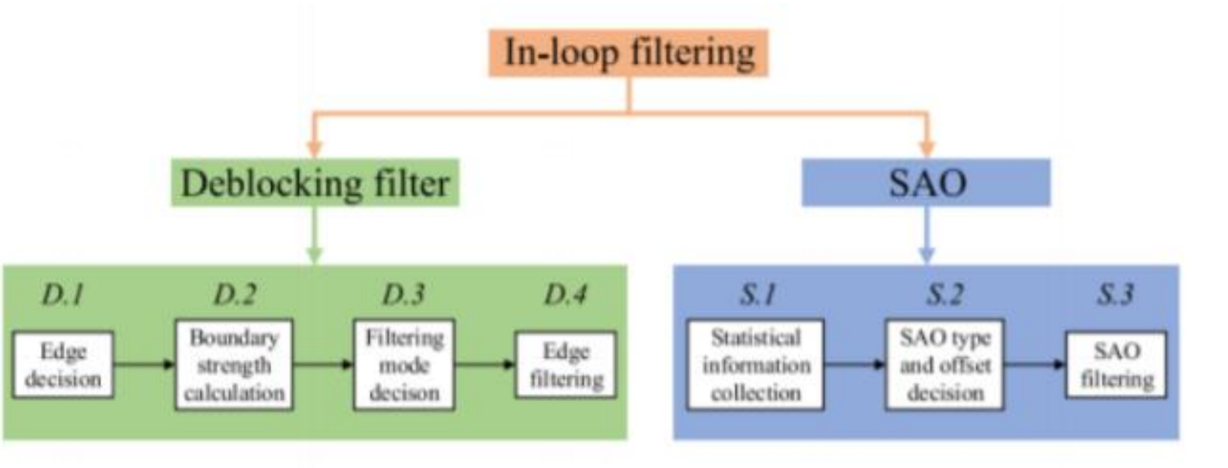


Figure 3.1: Loop Filtering With Deblocking and SAO[5]

Deblocking filter is applied to the edges aligned on 8*8 sample grids. As denoted in Fig-3.1, it can be split into four steps, namely edge decision, boundary strength calculation, filtering mode decision, and edge filtering. They are represented by D.1, D.2, D.3, and D.4 respectively. In D.1, only the edges on 8*8 grids, either prediction unit or transform unit boundaries, are considered to be filtered. In D.2, the boundary strength is calculated according to the coding information [2]. The boundary strength can take one of three possible values: 0, 1, and 2, which indicate no filtering, weak filtering, and strong filtering respectively In D.3, if the boundary strength value is 1 or 2, additional conditions are checked to determine whether the deblocking filtering should be applied. The weak or strong filtering mode is decided according to values of samples near the boundaries. In D.4, first vertical edges are filtered (horizontal filtering) then horizontal edges are filtered (vertical filtering) [2].

3.1.4 Sample Adaptive Offsets

As meant in Fig-3.1. , SAO comprises of three stages, specifically factual data gathering, SAO type and counterbalance choice, and SAO sifting. They are spoken to by S.1, S.2, and S.3 individually. In S.1, the quantity of pixels and the whole of mutilation of a certain SAO type are determined. In S.3, pixels in the CTU are sifted by including the comparing best counterbalanced qualities restrictively. The procedures of SAO in HEVC encoder and decoder are extraordinary [1].

For SAO in HEVC encoder, pixel-wise imperative exists in sort order and counterbalF ance calculation stages, which will bring down the parallelism of GPU. Because of this trouble, few existing calculations have been proposed to enhance SAO on GPU in HEVC encoder [3].

Chapter 4 : HEVC Standard Overview

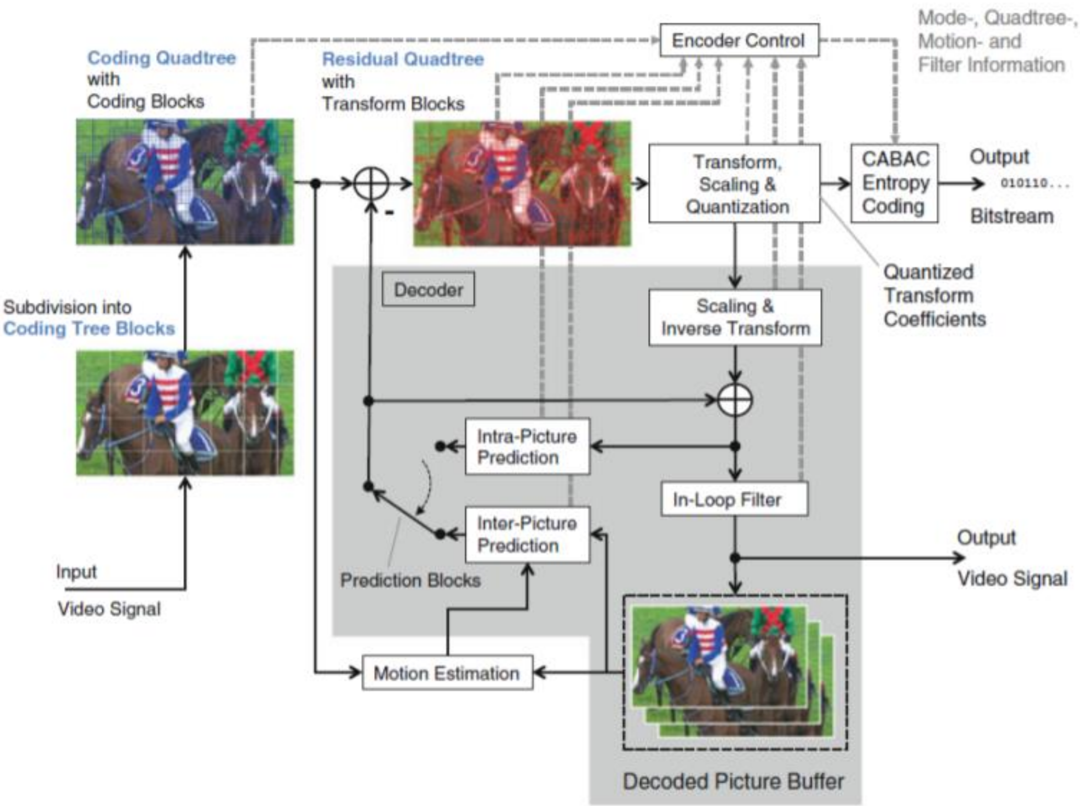


Figure 3.1: Block Diagram of High Efficiency Video Coding[5]

The plan of most video coding standards is fundamentally gone for having the most noteworthy coding efficiency. Coding efficiency is the capacity to encode video at the most minimal conceivable piece rate while keeping up a specific dimension of video qulity. There are two standard approaches to quantify the coding efficiency of a video coding standard, which are to utilize a goal metric, for example, peak signal-to-noise ratio (PSNR), or to utilize abstract evaluation of video quality. The Quality of video is viewed as the most imperative approach to quality a video coding standard since people see video quality abstractly.

HEVC profits by the utilization of bigger coding tree unit (CTU) sizes. This has been appeared in PSNR tests with a HM-8.0 HEVC encoder where it was compelled to utilize continuously littler CTU sizes. For all test successions, when contrasted and a 64*64 CTU estimate, it was demonstrated that the HEVC bit rate expanded by 2.2 percentage when compelled to utilize a 32*32 CTU measure, and expanded by 11.0 percentage when compelled to utilize a 16*16 CTU estimate. In the Class A test groupings, where the resolution of the video was 2560*1600, when contrasted and a 64*64 CTU estimate, it was demonstrated that the HEVC bit rate expanded by 5.7percentage when compelled to utilize a 32*32 CTU measure, and expanded by 28.2 percentage when compelled to utilize a 16*16 CTU estimate.

The tests demonstrated that extensive CTU sizes increment coding efficiency while likewise decreasing decoding time.As it is appeared in Figure 3.1, the HEVC design can be partitioned in two noteworthy parts, encoder and decoder. In the encoder part, every image is parceled in numerous units. In the subsequent stage, every unit is being anticipated by intra outline prediction or inter prediction and the outcome is subtracted from the first unit. The leftover is transformed for the most part by Discrete Cosine Transform (DCT) and quantized in the subsequent stage. Toward the end, transformed yield, prediction data, mode data and headers are entropy encoded. In decoder each partner of encoder squares does the turn around activity to convey the image to the opposite end of the correspondence.

In this area, We Initially propose the fundamental algorithm for quick CU measure choice,and after that we plan two valuable extra devices to upgrade its coding execution and to build time reduction, individually.The flowchat of our whole algorithm is attracted in Figure.

We see that the computational time decrease of low QP is littler than that of high QP. We additionally realize that the little estimated CUs are regularly utilized in low QP cases. The splitting choice happens regularly in the locale of grouped little estimated

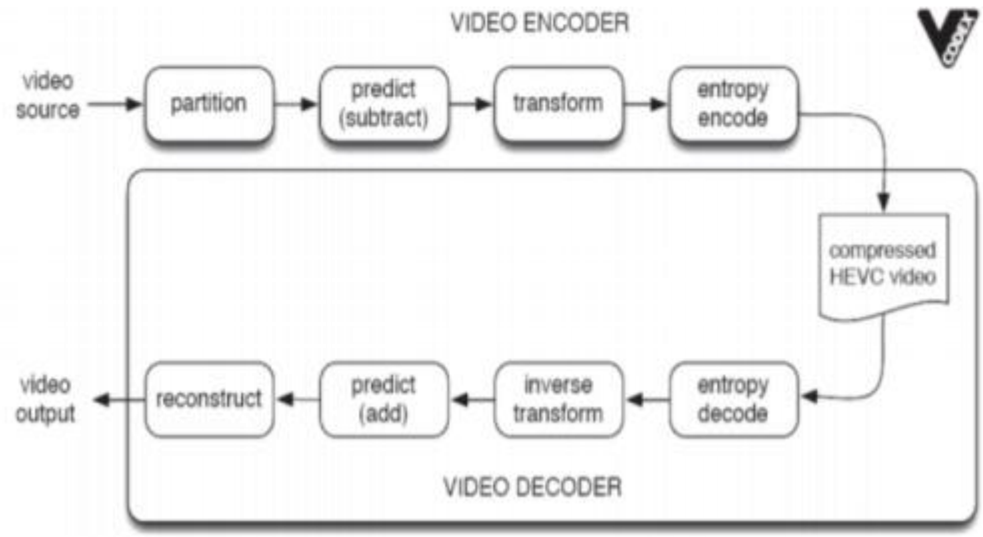


Figure 4.2: Structure of HEVC encoder and Decoder[8]

- **Encoder Steps**:

  The Encoder will perform following steps.

  1. The Encoder will segment one picture into numerous units.
  2. Then it will anticipate every unit with the utilization of inter prediction and intra prediction and substract prediction from unit.
  3. After that it will do Tranforming and quantizing the lingering that implies the diverse between unique picture and prediction.
  4. The encoding will tranform the yield , data of prediction , data of mode headers.

- **Decoder Steps**:

  The Decoder will perform following steps.

  1. Entropy decoding and extracting the contents of the encodede squence.
  2. In the next it will go for rescalling and inverting the transform picture.
  3. Then it will predict each unit of the picture and it will add the prediction to the output of the inverse trsansform.
  4. Then it will reconstruct decoded video picture.

CUs. Possibly, we can utilize just a single entomb prediction after the splitting choice to diminish the complexity further. There are two conceivable bury modes analyzed initially after the splitting decision,2N * N and N* 2N. We see that the shape is very reliant on the sizes of neighboring CUs. In the event that the quantity of little CU on the left-referenced CU is bigger than that on the best referenced CU, the encoder analyzes the RD cost of 2N * N at the present profundity. Else, we check just the N* 2N prediction. We test 8 HD groupings (64 outlines for each arrangement) in Table II to demonstrate the efficient enhancement by this 2N * N/N*2N pre - determination.

4.1 Transform, Scaling, and Quantization

From the encoder viewpoint, change coding comprises of three progressive stages: the change itself, the quantization, lastly the entropy coding of quantization files related with the quantized change coefficients, likewise indicated as change coefficient levels. This segment manages the entropy coding some portion of change coding in H.265/HEVC.

As opposed to H.264/MPEG-4 AVC, H.265/HEVC indicates setting based versatile context based adaptive binary arithmetic coding (CABAC) as the main technique for entropy coding. CABAC in H.265/HEVC acquires from its partner in H.264/MPEG-4 AVC. The three essential strides of preparing: binarization, setting demonstrating, and binary arithmetic coding (BAC). The BAC motor of H.264/MPEG-4 AVC, otherwise called M coder, including likelihood estimation and low-multifaceted nature sidestep coding mode, is reused for H.265/HEVC with no adjustment. In any case, the solid manifestations of binarization and setting displaying in H.265/HEVC are modified significantly in respect to what is indicated in H.264/MPEG-4 AVC. Accordingly, the change coefficient level coding in H.265/HEVC is altogether enhanced, both as for throughput and coding effectiveness.

4.2 Core Transform

Two-dimensional changes are figured by applying 1-D changes in the flat and vertical bearings. The components of the centre change frameworks were inferred by apraxia mating scaled DCT premise capacities, under contemplations, for example, restricting the fundamental powerful range for change calculation and amplifying the accuracy and closeness to symmetry when the grid sections are indicated as whole number qualities.

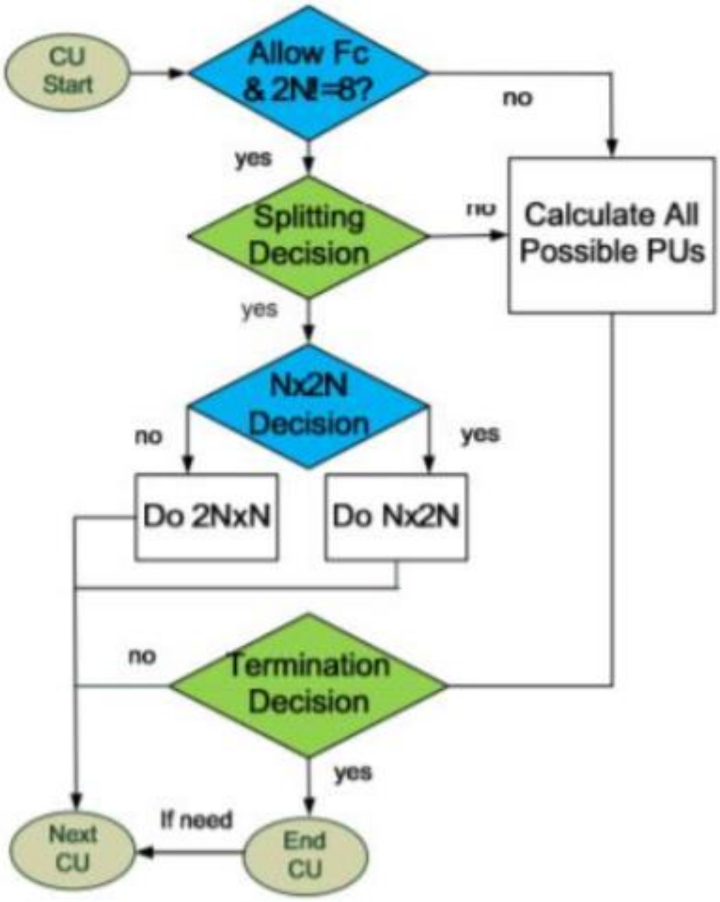


Figure 4.3: Proposed algorithm flowchart for processing a Coding Unit[6]

For instance, the lattice for the length-16 change is as appeared in the condition at the base of the past page. The lattices for the length-8 and length-4 changes can be inferred by utilizing the initial eight passages of columns 0, 2, 4. . . And utilizing the initial four sections of columns 0, 4, 8. . . Individually. Despite the fact that the standard indicates the change basically as far as the estimation of a lattice, the estimations of the sections in the network were chosen to have key symmetry properties that empower quick incompletely considered executions with far less numerical activities than a normal framework duplication, and the bigger changes can be built by utilizing the littler changes as building squares.

4.3 Scaling and Quantization

The transform matrix are close approximations of estimations of consistently scaled premise elements of the orthonormal DCT, the prescaling activity that is fused in the dep quantization of H.264/MPEG-4 AVC isn't required in HEVC. This shirking of recurrence explicit premise work scaling is valuable in decreasing the middle of the road memory estimate, particularly while thinking about that the span of the change can be as huge as 32*32. For quantization, HEVC utilizes basically the equivalent URQ plot controlled by a quantization parameter (QP) as in H.264/MPEG-4 AVC. The scope of the QP esteems is characterized from 0 to 51, and an expansion by 6 copies the quantization step size to such an extent that the mapping of QP qualities to step sizes is roughly logarithmic.Quantization scaling matrices are likewise bolstered.

To diminish the memory expected to store frequency-specific scaling esteems, just quantization networks of sizes 4*4 and 8*8 are utilized. For the bigger changes of 16*16 and 32*32 sizes, a 8*8 scaling lattice is sent and is connected by sharing qualities inside 2*2 and 4*4 coefficient bunches in recurrence subspacesaside from qualities at DC (zero recurrence) positions, for which particular qualities are sent and connected.

Superior and equipment productive models for the acknowledgment of the HEVC quantization systems is exhibited. The proposed structures can be utilized not excluq sively to acknowledge HEVC quantizers and de-quantizers yet additionally brought tos gether quantization circuits for the calculation of the two strategies with diminished equipment cost. Furthermore, they can be effectively designed to give executions offering diverse exchange offs between execution, idleness and equipment cost, making them very appropriate for different application spaces with particular necessities.

Chapter 5 : HEVC ENCODER AND DECODER

5.1 : Coding Efficiency

The coding eeficiency of video is increase on high point with the help of hevc encoder. In addition the coding efficiency is the capable to encode video at the low possible bit rate and it will maintain video quality too.There are two important methods calculate video coding standard , which are using in to objective metric like perl ration to noise ratio, and to use of subjective assesment of coding efficiency.which would be consider as the most important method to measure video coding as well as maintain video quality.

The use of PNSR tests with a HM-8.0 HEVC encoder is forcly use the smaller CTU Sizes. For the experiment result to compare the results to a 64*64 CTU size , it was show that the bitrate of the HEVC is increased with the 2.2% when forced to use the 32*32 size then it was shwoing the HEVC is increased by 11.0% when it will forced to use 16*16 CTU sizes.where the resolution of the video is 4k resolution and 8k resolution.for the 4k resolution would be suppported by HEVC and it can use the different sizes of the CTU and to comparing to the 64*64 CTU size , it shows the bitrate of HEVC is increased by 5.7% when forcing to use the 32*32 CTU size in addition it shows 28% of the bitrate of the HEVC application with the use of 16*16 CTU size.The end of the experiment I can see the large size of the CTU is increase the coding efficiency and also it is reducing the time of the decoding.

5.2 : Coding Tools

5.2.1:Coding Tree Unit:

Previously , HEVC used 16*16 pixel macro blocks which is replace by HEVC with the 64*64 samples of the squares structures. In addition to that It can do better suballocate pictures in components measured structures.The main work of HEVC is improving the coding efficiency which can devide one picture into different sizes of CTUs like as 16*16 ,32*32 and 64*62 with biger picture bit sizes.

5.2.2: Parallel Handling tools:

The picture to be divided into grid of rectangular regions that is indivisually decoded and encoded.The parallel processing is allow to divide into tiles and that is the main advantage of the tiles.and one tile can be indivisually decoded and tile can allow for the random access Parallel Proccesing WPP are allowed tiles. If tiles are presents , and they must be raise upto 64 pixels 256 wide including the level of specific linmits on the number which tiles allowed. The slices can report the data with the decoded part which are independentenly from each other in the main purpose of tiles being the resyncronization if the data lost in the video stream.

The slices can be the data defined as self-contained in that prediction is not making for the data across the slice boundaries. When it will go to the loop filtering done on the picture done to the data and information across slice boundaries.avarage of the data correction.

How to build this Application:

- Requirements:

  1. GCC 5.4.0 or later
  2. CMake 3.5.1 or later
  3. YASM Assembler version 1.2.0 or later

- Instruction for build the applocation:

  1. In the main repository, run: mkdir build && cd build && cmake .. && make -j $(nproc) && sudo make install

- Installation Steps for Application:

  1. On any of the Linux* Operating Systems listed above, copy the binaries under a location of your choice.
  2. Change the permissions on the sample application "SvtHevcEncApp" executable by running the command: chmod +x SvtHevcEncApp
  3. cd into your chosen location
  4. Run the sample application to encode. ./SvtHevcEncApp -i [in.yuv] -w [width] -h [height] -b [out.265]
  5. Sample application supports reading from pipe. E.g. ffmpeg -i [input.mp4] -nostdin -f rawvideo -pix_fmt yuv420p - | ./SvtHevcEncApp -i stdin -n [number_of_frames_to_encode] -w [width]

- Sample Command of the Application.

  ./SvtHevcEncApp -w 4096 -h 2160 -i /home/pg3/content/Netflix_TunnelFlag_4096x2160_10bit_unpacked_60Hz_P420_2bitspack.yuv -q 30 -fps 60 -n 5000 -intra-period 55 -bit-depth 10 -c Real-timeEncodingMode.cfg -stat-report 0 -lad 17 -hdr 0 -compressed-ten-bit-format 1 -nch 1 -nb 191 -b output.265 -asm 1 –encMode

- Parameters in the Application which we can use:

-nch	NumberOfChannels in the video.
-i	InputFile of the video
-b	StreamFile of the video
-errlog	ErrorFile
-o	ReconFile
-qp-file	QpFile
-interlaced-video	InterlacedVideo
-separate-fields	SeperateFields
-w	SourceWidth
-h	SourceHeight
-n	FrameToBeEncoded
-nb	BufferedInput
-base-layer-switch-mode	BaseLayerSwitchMode

-encMode	EncoderMode
-intra-period	IntraPeriod
-irefresh-type	IntraRefreshType
-fps	FrameRate
-fps-num	FrameRateNumerator
-fps-denom	FrameRateDenominator
-bit-depth	EncoderBitDepth
-color-format	EncoderColorFormat
-compressed-ten-bit-format	CompressedTenBitFormat
-hierarchical-levels	HierarchicalLevels
-pred-struct	PredStructure
-scd	SceneChangeDetection
-q	QP
-use-q-file	UseQpFile
-rc	RateControlMode
-lad	LookAheadDistance
-tbr	TargetBitRate
-max-qp	MaxQpAllowed
-min-qp	MinQpAllowed
-dlf	LoopFilterDisable
-sao	SAO
-use-default-me-hme	UseDefaultMeHme
-hme	HME
-search-w	SearchAreaWidth
search-h	SearchAreaHeight
-constrd-intra	ConstrainedIntra
-tune	Tune
-lp	LogicalProcessors
-ss	TargetSocket
-rt	SwitchThreadsToRtPriority
-brr	BitRateReduction
-sharp	ImproveSharpness
-vid-info	VideoUsabilityInfo
-hdr	HighDynamicRangeInput
-ua-delm	AccessUnitDelimiter
-pbuff	BufferingPeriod
-tpic	PictureTiming
-reg-user-data	RegisteredUserData
-unreg-user-data	UnregisteredUserData
-recovery-point	RecoveryPoint
-max-cll	MaxCLL
-max-fall	MaxFALL
-use-master-display	UseMasterDisplay
-master-display	MasterDisplay
-level	Level
-asm	AsmType

Encoder Parameter as shown in the configuration file	Command Line parameter	Range	Default	Description
	-nch	[1 - 6]	1	Number of encode instances
	-c	any string	null	Configuration file path
Input File	-i	any string	None	Input file path
StreamFile	-b	any string	null	output bitstream file path
ErrorFile	-errlog	any string	stderr	error log displaying configuration or encode errors
UseQpFile	-use-q-file	[0 - 1]	0	When set to 1, overwrite the picture qp assignment using qp values in QpFile
QpFile	-qp-file	any string	Null	Path to qp file
EncoderMode	-encMode	[1 - 6]	2	A preset defining the quality vs density tradeoff point that the encoding is to be performed at. (e.g. 1 is the highest quality mode, 6 is the highest density mode)
LatencyMode	-latency-mode	[0 - 1]	0	For lower latency (0: Normal Latency, 1: Low Latency)
EncoderBitDepth	-bit-depth	[8 , 10]	8	specifies the bit depth of the input video
CompressedTenBitFormat	-compressed- ten-bit-format	[0 - 1]	0	Offline packing of the 2bits: requires two bits packed input (0: OFF, 1: ON)

FrameToBeEncoded	-n	[0 - 2^31 - 1]	0	Number of frames to be encoded, if number of frames is > number of frames in file, the encoder will loop to the beginning and continue the Encode. Use -1 to not buffer.
BufferedInput	-nb	[-1, 1 to 2^31 -1]	-1	number of frames to preload to the RAM before the start of the encode If -nb = 100 and –n 1000 --> the encoder will encode the first 100 frames of the video 10 times
Profile	-profile	[1 - 2]	2	1: Main, 2: Main 10
Tier	-tier	[0 - 1]	0	0: Main, 1: High
Level	-level	[1, 2, 2.1,3, 3.1, 4, 4.1, 5, 5.1, 5.2, 6, 6.1, 6.2]	0	0 to 6.2 [0 for auto determine Level]
FrameRate	-fps	[0 - 2^64 - 1]	25	If the number is less than 1000, the input frame rate is an integer number between 1 and 60, else the input number is in Q16 format (shifted by 16 bits) [Max allowed is 240 fps]
FrameRateNumerator	-fps-num	[0 - 2^64 - 1]	0	Frame rate numerator e.g. 6000
FrameRateDenominator	-fps-denom	[0 - 2^64 - 1]	0	Frame rate denominator e.g. 100
Injector	-inj	[0 - 1]	0	Enable injection of input frames at the specified framerate (0: OFF, 1: ON)
InjectorFrameRate	-inj-frm-rt	[1 - 240]	60	Frame Rate used for the injector. Recommended to match the encoder speed.

SpeedControlFlag	-speed-ctrl	[0 - 1]	0	Enables the Speed Control functionality to achieve the real-time encoding speed defined by –fps. When this parameter is set to 1 it forces –inj to be 1

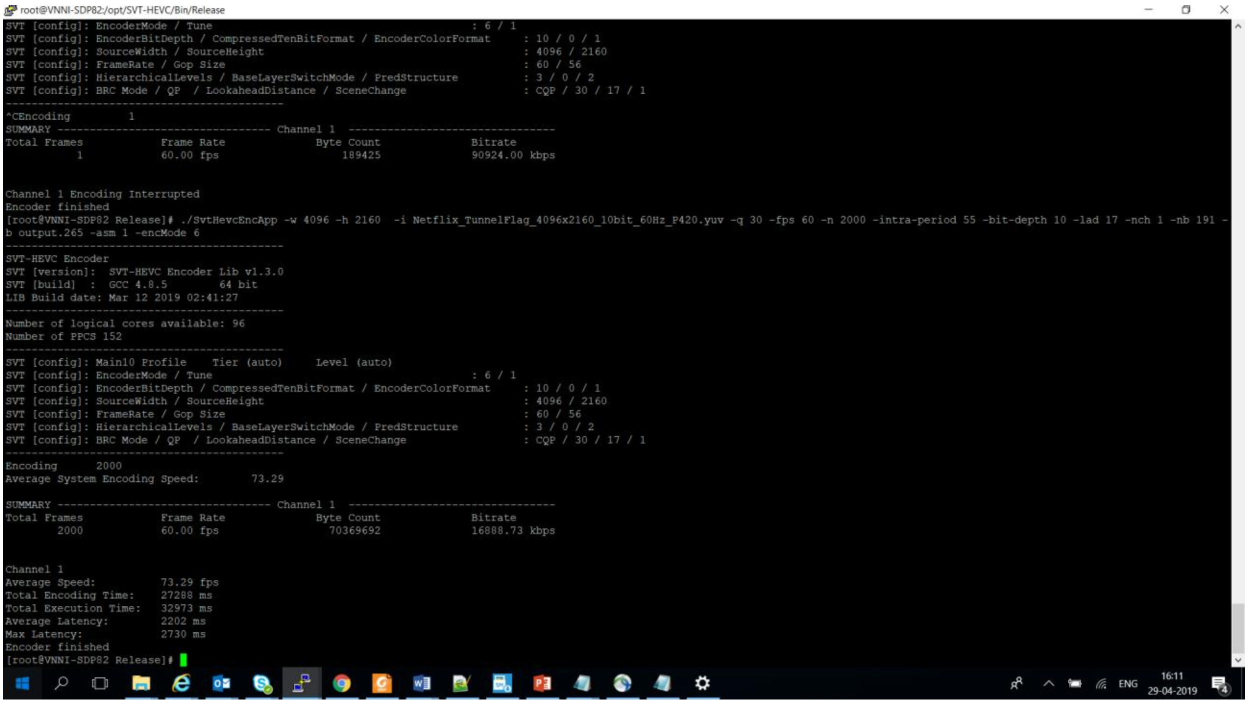
- Sample Output of the Application.

```
SVT-HEVC Encoder
SVT [version]:  SVT-HEVC Encoder Lib v1.3.0
SVT [build]  :  GCC 4.8.5       64 bit
LIB Build date: Mar 12 2019 02:41:27
-------------------------------------------
Number of logical cores available: 96
Number of PPCS 152
-------------------------------------------
SVT [config]: Main10 Profile    Tier (auto)     Level (auto)
SVT [config]: EncoderMode / Tune                                                    : 6 / 1
SVT [config]: EncoderBitDepth / CompressedTenBitFormat / EncoderColorFormat : 10
SVT [config]: SourceWidth / SourceHeight                                            : 4096/2160
SVT [config]: FrameRate / Gop Size                                                  : 60 / 56
SVT [config]: HierarchicalLevels / BaseLayerSwitchMode / PredStructure   : 3 / 0 / 2
SVT [config]: BRC Mode / QP  / LookaheadDistance / SceneChange           : 30/ 17 / 1
-------------------------------------------
Encoding     2000
Average      System      Encoding      Speed:                          70.28
4000
Average      System      Encoding      Speed:                          72.86
5000
SUMMARY --------------------------------- Channel 1  --------------------------------
Total Frames          Frame Rate           Byte Count           Bitrate
      5000         60.00 fps           175810086          16877.77 kbps



Channel 1
Average Speed:            74.79 fps
Total Encoding Time:   66850 ms
Total Execution Time:  75879 ms
Average Latency:          2259 ms
Max Latency:              2843 ms
```

- **Screenshots of the outputs:**

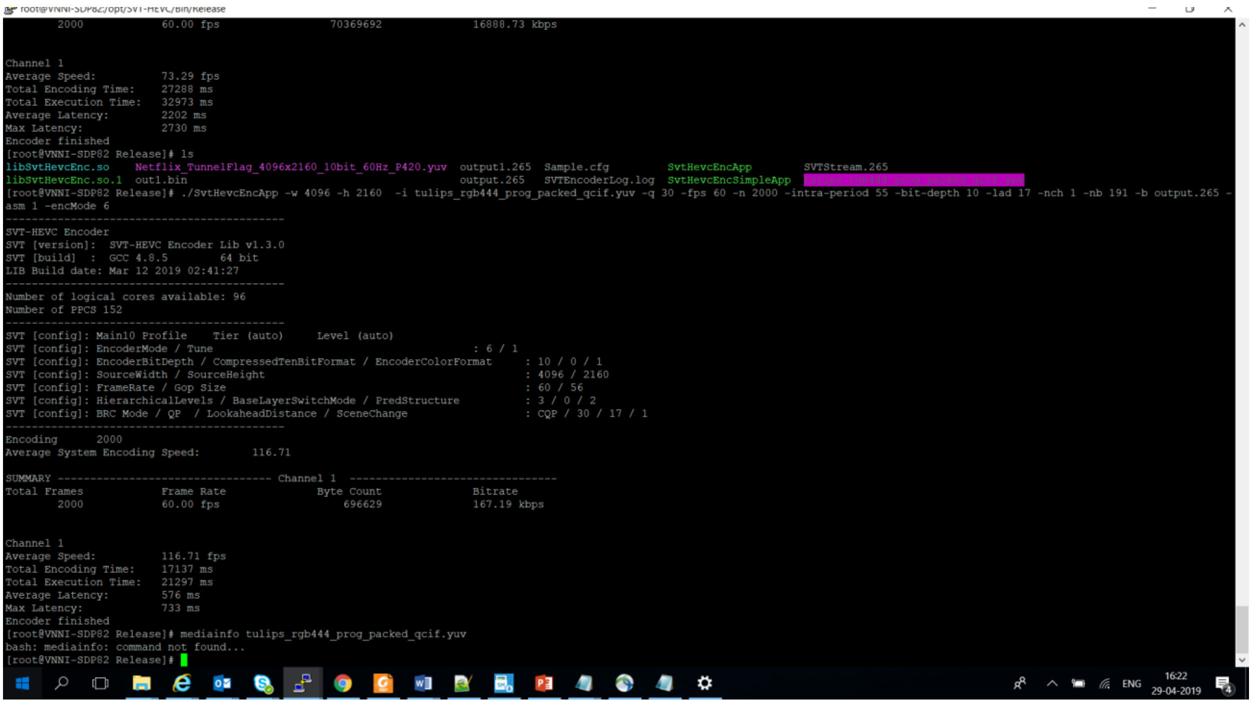

root@VNNI-SDP82:/opt/SVT-HEVC/Bin/Release

```
SVT [config]: EncoderMode / Tune                                          : 6 / 1
SVT [config]: EncoderBitDepth / CompressedTenBitFormat / EncoderColorFormat     : 10 / 0 / 1
SVT [config]: SourceWidth / SourceHeight                                  : 4096 / 2160
SVT [config]: FrameRate / Gop Size                                        : 60 / 56
SVT [config]: HierarchicalLevels / BaseLayerSwitchMode / PredStructure    : 3 / 0 / 2
SVT [config]: BRC Mode / QP  / LookaheadDistance / SceneChange            : CQP / 30 / 17 / 1
-------------------------------------------
^CEncoding         1
SUMMARY --------------------------------- Channel 1  --------------------------------
Total Frames            Frame Rate              Byte Count              Bitrate
           1            60.00 fps                  189425               90924.00 kbps


Channel 1 Encoding Interrupted
Encoder finished
[root@VNNI-SDP82 Release]# ./SvtHevcEncApp -w 4096 -h 2160  -i Netflix_TunnelFlag_4096x2160_10bit_60Hz_P420.yuv -q 30 -fps 60 -n 2000 -intra-period 55 -bit-depth 10 -lad 17 -nch 1 -nb 191 -
b output.265 -asm 1 -encMode 6
-------------------------------------------
SVT-HEVC Encoder
SVT [version]:  SVT-HEVC Encoder Lib v1.3.0
SVT [build]  :  GCC 4.8.5        64 bit
LIB Build date: Mar 12 2019 02:41:27
-------------------------------------------
Number of logical cores available: 96
Number of PPCS 152
-------------------------------------------
SVT [config]: Main10 Profile    Tier (auto)     Level (auto)
SVT [config]: EncoderMode / Tune                                          : 6 / 1
SVT [config]: EncoderBitDepth / CompressedTenBitFormat / EncoderColorFormat     : 10 / 0 / 1
SVT [config]: SourceWidth / SourceHeight                                  : 4096 / 2160
SVT [config]: FrameRate / Gop Size                                        : 60 / 56
SVT [config]: HierarchicalLevels / BaseLayerSwitchMode / PredStructure    : 3 / 0 / 2
SVT [config]: BRC Mode / QP  / LookaheadDistance / SceneChange            : CQP / 30 / 17 / 1
-------------------------------------------
Encoding      2000
Average System Encoding Speed:        73.29

SUMMARY --------------------------------- Channel 1  --------------------------------
Total Frames            Frame Rate              Byte Count              Bitrate
        2000            60.00 fps                70369692               16888.73 kbps


Channel 1
Average Speed:          73.29 fps
Total Encoding Time:    27288 ms
Total Execution Time:   32973 ms
Average Latency:        2202 ms
Max Latency:            2730 ms
Encoder finished
[root@VNNI-SDP82 Release]#
```


root@VNNI-SDP82:/opt/SVT-HEVC/Bin/Release
2000 60.00 fps 70369692 16888.73 kbps
Channel 1
Average Speed: 73.29 fps
Total Encoding Time: 27288 ms
Total Execution Time: 32973 ms
Average Latency: 2202 ms
Max Latency: 2730 ms
Encoder finished
[root@VNNI-SDP82 Release]# ls
libSvtHevcEnc.so Netflix_TunnelFlag_4096x2160_10bit_60Hz_P420.yuv output1.265 Sample.cfg SvtHevcEncApp SVTStream.265
libSvtHevcEnc.so.1 out1.bin output.265 SVTEncoderLog.log SvtHevcEncSimpleApp tulips_rgb444_prog_packed_qcif.yuv
[root@VNNI-SDP82 Release]# ./SvtHevcEncApp -w 4096 -h 2160 -i tulips_rgb444_prog_packed_qcif.yuv -q 30 -fps 60 -n 2000 -intra-period 55 -bit-depth 10 -lad 17 -nch 1 -nb 191 -b output.265 -asm 1 -encMode 6

SVT-HEVC Encoder
SVT [version]: SVT-HEVC Encoder Lib v1.3.0
SVT [build] : GCC 4.8.5 64 bit
LIB Build date: Mar 12 2019 02:41:27

Number of logical cores available: 96
Number of PPCS 152

SVT [config]: Main10 Profile Tier (auto) Level (auto)
SVT [config]: EncoderMode / Tune : 6 / 1
SVT [config]: EncoderBitDepth / CompressedTenBitFormat / EncoderColorFormat : 10 / 0 / 1
SVT [config]: SourceWidth / SourceHeight : 4096 / 2160
SVT [config]: FrameRate / Gop Size : 60 / 56
SVT [config]: HierarchicalLevels / BaseLayerSwitchMode / PredStructure : 3 / 0 / 2
SVT [config]: BRC Mode / QP / LookaheadDistance / SceneChange : CQP / 30 / 17 / 1

Encoding 2000
Average System Encoding Speed: 116.71
SUMMARY -------------------------------- Channel 1 --------------------------------
Total Frames Frame Rate Byte Count Bitrate
2000 60.00 fps 696629 167.19 kbps
Channel 1
Average Speed: 116.71 fps
Total Encoding Time: 17137 ms
Total Execution Time: 21297 ms
Average Latency: 576 ms
Max Latency: 733 ms
Encoder finished
[root@VNNI-SDP82 Release]# mediainfo tulips_rgb444_prog_packed_qcif.yuv
bash: mediainfo: command not found...
[root@VNNI-SDP82 Release]#
ENG
16:22
29-04-2019

Chapter 6 : Experimental Results

In the result of High efficiency video coding there is the input parameters are as resolution and the results are pretty good and incremental. I collected result for speed with bitrate and Encoding speed vs CPU utilization. You can see the graph results that showing that could bitrate is increase with speed. We can see the results data with increment of the performance.

No. of Frames	Speed/FPS	Bitrate/kbps
1000	147.99	1285.53
2000	155.64	1285.55
3000	158.6	1285.55
4000	159.87	1285.56
5000	160.52	1285.56
6000	161.22	1285.55
7000	161.31	1276

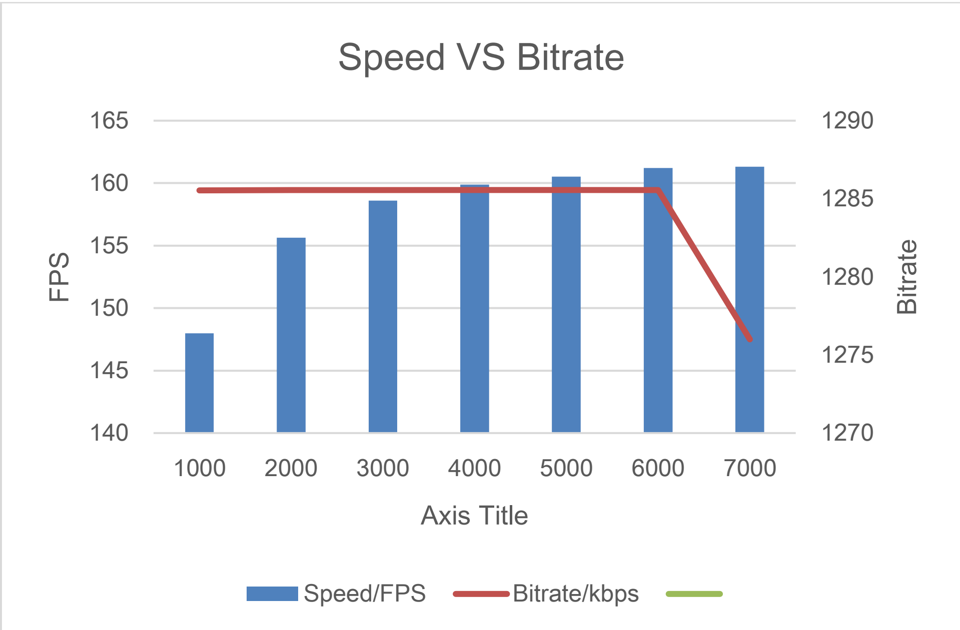


Figure 5.1 : Encoding Speed vs Bitrate

No. of CPU	No. of Frames	Speed/fps	Latency /ms
4	5000	21.33	7695
8	5000	37.99	4250
16	5000	64.19	2508
32	5000	114.89	1471
64	5000	153.43	1072
96	5000	160.89	1002

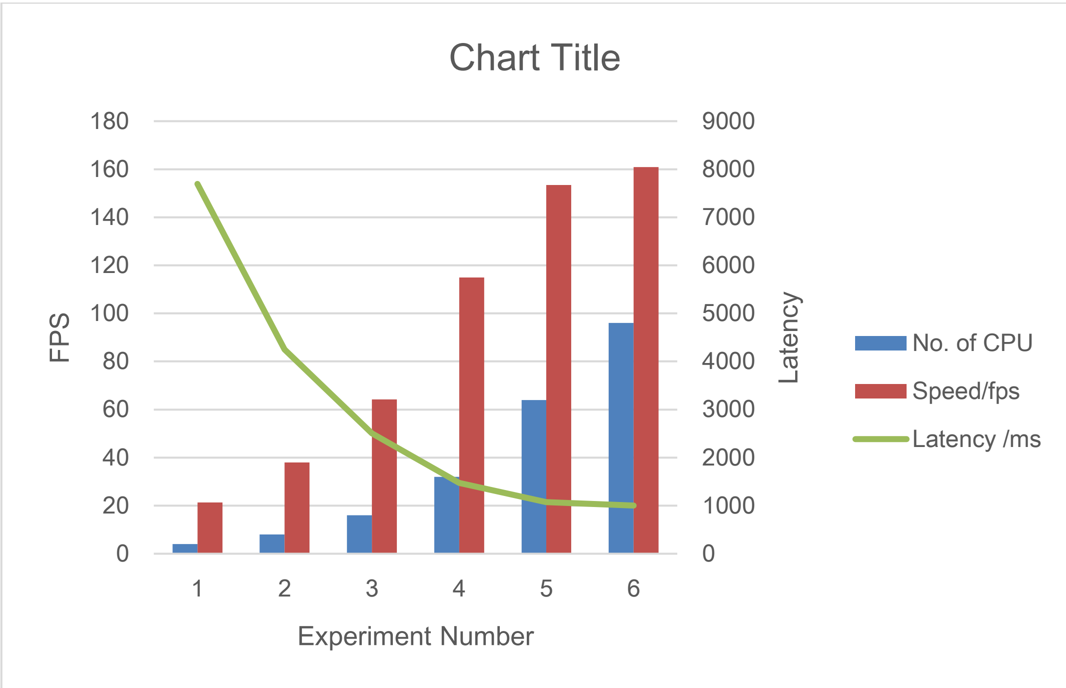


Figure 5.2 : Encoding Speed vs CPU

Conclusion

The outcomes recorded with this report is system performance is increase with the use of CPU scaling in the commands which is not there in the previous standard. H.264/MPEG-4 AVC don't have that much efficiency of the coding which is problem with coding standard. Now we have increased the performance 45% according to Advanced Vector Coding. The Results presents the value of latency is decrease and the speed of the encoding standard is increase which is useful for the HEVC standard.

Bibliography

1. Gary J. Sullivan, Fellow, IEEE, Jens-Rainer Ohm, Member, IEEE, Woo-Jin Han,Member, IEEE, and Thomas Wiegand, Fellow, IEEE
2. Yang Wang , Xun Guo, Member, IEEE, Xiaopeng Fan , Senior Member, IEEE, Yan Lu, Debin Zhao , Member, IEEE, and Wen Gao, Fellow, IEEE

3. Stefan Radicke1,2, Jens-Uwe Hahn1, Christos Grecos2 and Qi Wang2 1Hochschule derMedien, Nobelstrae 10, Stuttgart, Germany 2School of Computing, University of the West of Scotland, High Street, Paisley, U.K. Radicke,Hahn@HdM-Stuttgart.de,Christos.Grecos, Qi.Wang@UWS.ac.uk
4. Dan Grois, Senior Member, IEEE, Tung Nguyen, Detlev Marpe, Fellow, IEEE Video Coding & Analytics Department, Image & Video Coding Group, Fraunhofer Heinrich Hertz Institute (HHI), Berlin, Germany E-mail: grois@ieee.org, {tung.nguyen,detlev.marpe}@hhi.fraunhofer.de
5. Yeong-Kang Lai, Senior Member, IEEE and Lee-Sheng Lien, Member, IEEE Department of the Electrical Engineering, National Chung Hsing University, Taichung 402, Taiwan, ROC Yeong-Kang Lai:yklai@dragon.nchu.edu.tw and Lee-Sheng Lien: w104064030@mail.nchu.edu.tw
6. Jens-Rainer Ohm, Member, IEEE, Gary J. Sullivan, Fellow, IEEE, Heiko Schwarz,Thiow Keng Tan, Senior Member, IEEE, and Thomas Wiegand, Fellow, IEEE.
7. Jung-Ah Choi and Yo-Sung Ho Gwangju Institute of Science and Technology (GIST) Gwangju, Republic of Korea {jachoi, hoyo}@gist.ac.kr
8. Abdelrahman AbdelazimA, Ahmed M. HamzaB, Djamel Ait-BoudaoudB American University of the Middle East Kuwait City, Kuwait University of Portsmouth, Portsmouth, U
9. Subhash Panwar Dept. of Computer Science and Engg.Govt. Engineering College Bikaner Bikaner, India ,e-mail: panwar.subhash@gmail.com
10. Chang-Yi Hong, Hui-Jane Hsieh* and Yinyi Lin Department of Communication Engineering National Central University, Taiwan Department of Information Management Chien Hsin University of Science and Technology, Taiwan yilin@ce.ncu.edu.tw
11. Pao-Chi Chang National Central University, Department of Communication Engineering No.300, Jhongda Rd., Taoyuan City, Taiwan, 320 ,e-mail: pcchang@ce.ncu.edu.tw
12. Gary J. Sullivan, Fellow, IEEE, Jens-Rainer Ohm, Member, IEEE, Woo-Jin Han, Member, IEEE, and Thomas Wiegand, Fellow, IEEE.