International Journal of Hybrid Information Technology
Vol.8, No.7 (2015), pp.319-334
http://dx.doi.org/10.14257/ijhit.2015.8.7.29

Analysis of HEVC Coding Standard

[1]Jianjun Wang[a] and [3]Li Zhou[b*], and [3]Tao Sun

[1,2]School of Information Science and Engineering, Shandong University, Jinan, Shandong, China
[3]Shandong Provincial Key Laboratory of Network based Intelligent Computing, University of Jinan, Jinan, China
*[a]henrywangjj@gmail.com, [b]zhou_Li@sdu.edu.cn, ise_sunt@ujn.edu.cn [*Corresponding Author]*

Abstract

HEVC video coding standard can achieve an average of 50% bit-rate saving, transmitting nearly the same visual quality, compared with the H.264/AVC High Profile. This paper focuses on the key features of the HEVC standard. Then it analyzes the complexity of the HEVC standard by profiling the HEVC reference software HM12.0 which is released by JCT-VC community with Intel Core i7 processor. The experiment covers different settings for a wide variety of video sequences, with the resolutions ranging from 240p to 1600p. The evaluation result shows that the HEVC standard can achieve a doubling in coding efficiency at the cost of increasing the complexity slightly. When the sequences are coded in all-intra or random-access configurations, intra prediction, loop filter, inverse transform and motion compensation are the most complex components for HEVC decoder. The motion compensation part should be given lots of consideration since it can take nearly half of the decoding time. The quantization parameter (QP) also has obvious influences on the complexity of the decoder. Besides, in practice, the HM decoder is not able to decode video content with 1080p or higher resolutions for real-time applications.

Keywords: *High efficiency video coding (HEVC), HEVC Test Model (HM), coding standard, H.264*

1. Introduction

As the resolution of the video sequences grows higher and higher, it's necessary to provide more sophisticated video compression technologies with higher coding efficiency than the previous video coding standard. Therefore, the Joint Collaborative Team on Video Coding (JCT-VC) releases the HEVC video coding standard as a successor to the previous H.264/AVC standard. The first edition of the HEVC standard has been finalized, with an aligned text published by both ITU-T and ISO/IEC in April, 2013. The HEVC standard is able to get a doubling of the coding efficiency compared with the H.264/AVC High Profile, transmitting same or higher video quality with an average of 50% bit-rate saving [1].

This paper mainly focuses on the detailed analysis of the highlighted features of the HEVC video coding standard by profiling the HEVC reference software. Section II analyzes the new features of HEVC, with which the new standard is able to achieve the demanded bit-rate saving and complexity-reducing compared to the previous H.264/AVC video coding standard. The readers can refer to the first edition of the finalized HEVC standard [2] or an overview of HEVC standard [3] to get a better overview of the new standard.

The rest parts of the paper cover mainly on the evaluation and profiling of the HEVC reference software (HM). The HM, with both the encoder and decoded

ISSN: 1738-9968 IJHIT

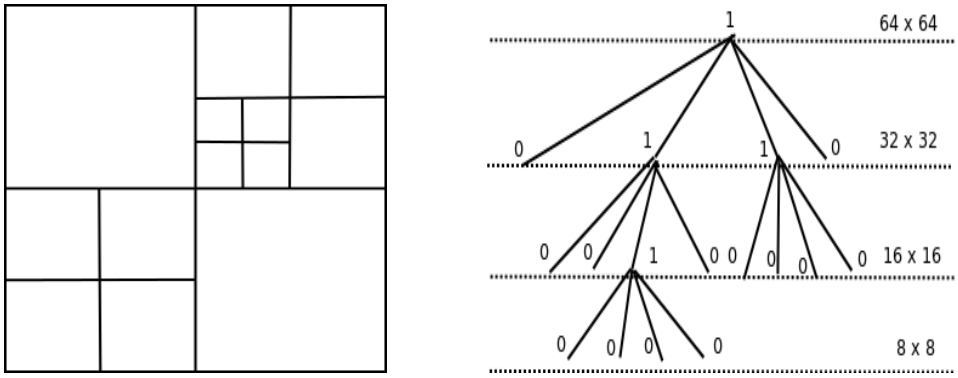

Figure 1. Example Of CTU Partitioning When The Size Of CTU Is Equal To 64 X 46 And The Maximum Depth Of The Partition Is 4. (A) CTU Partitioning (B) Corresponding Quadtree Structure

included, written in C++, released by JCT-VC community, serves as a reference implementation of the HEVC video coding standard. The version of HM chosen by this paper is 12.0, since it corresponds to the first edition of the finalized specification. Section III illustrates the encoder and the decoder of the HEVC video coding standard, with both the architecture and a brief description of each component of the encoder or decoder provided. Section IV gives the detailed experimental results by evaluating the reference software by profiling the HM with different configurations. The experiment uses the sequences recommended by JCT-VC, and the resolutions ranges 240p to 1600p. Configurations include all-intra (AI), random-access (RA), low-delay (LD), and different quantization parameters (QP) including 22, 27, 32 and 37 are used. Not only the experimental results, but a deep analysis is given in this section. At the end of this paper, Section V gives a brief conclusion of the paper.

2. Key Features of the HEVC Standard

2.1. More Flexible Block Partitioning and Bigger Blocks

In video coding, a picture is usually divided into blocks to reduce the coding complexity. In H.264/AVC, a 16x16 block partition, called macro-block (MB), is used to serve as the basic unit when coding or decoding the original videos. However, due to the ever-increasing resolutions of the video, partitions with fixed size of small blocks usually can not get the demanded coding efficiency since partitions with fixed size of small blocks are not sufficient to capture the increased spatial correlation in higher resolution content [4-8]. In HEVC, a more flexible block partitioning and bigger blocks are introduced. In HEVC, the coding tree unit (CTU), instead of the Macro Block (MB), lies in the core of the video coding layer, serving as the basic unit when processing video contents with HEVC standard. The "unit" in HEVC encapsulates the block, which contains the arrays of content-specified data, and the associated syntax structure [9]. The largest CTU can be up to 64x64 pixels, and each CTU can be subdivided using a recursive quad-tree structure until the leaves are reached. The leaf of a CTU is called the coding unit (CU), which is the basic unit of the coding or decoding process. One CU consists of the corresponding coding blocks (CB), the associated syntax structures, the transform unit (TU) and the prediction unit (PU). Fig. 1 illustrates an example of the CTU partitioning and the corresponding quad tree structure. This bigger and more

flexible quad-tree partitioning has three obvious advantages compared to the previous H.264/AVC standard. It's proved that bigger block partitioning could increase coding efficiency significantly for video sequences with higher resolutions. It leaves more freedom for the encoder to choose the size and depth of the partition according to the different resolutions and type of the video content. For example, for a 1080p content, which may contain complex motion activities of small regions, a CTU size of 64 with maximum depth of 4 would be more preferable, while for a CIF (352x288 pixels) or a 1080p (1920x1080 pixels) content with less motion activities, a CTU size of 16 or 64 with maximum depth of 2 would be more suitable. With only CU provided as the basic unit, by removing the distinction between macro-block and sub-macro-block, the quad-tree structure can be specified and processed in a very simple way.

2.2. Improved Mechanisms to Support Parallel Processing

On one hand, the coding efficiency of HEVC has been increased; on the other hand, the complexity is increased. In HEVC, two special methods are used to support parallel processing to improve the throughput.

1) Tile-based processing: In H.264/AVC standard, slices, which contain a collection of macro blocks, are used to support parallel processing. Each slice can be independently decoded, without using information from any other slices. However, lower coding efficiency and increased complexity are caused because of disabling the prediction across slice boundaries. In HEVC, except for the slices, tiles, which are a rectangular region of CTUs, are introduced to support parallel processing. Similar to slices, each tile can be independently decoded, without using information from any other tiles, and the complexity is lower than what's caused by slices.
2) Wavefront parallel processing: To avoid disabling the prediction across slice or tile boundaries, HEVC introduced a new method called wavefront parallel process (WPP) to support parallel processing. In this method, the CTUs of a frame are arranged in different rows. The first row is processed normally, and the next row is processed when the first two CTUs of the previous row have been decoded.

2.3. More Intra Prediction Modes for Intra Prediction

In HEVC, intra-prediction operates differently according to the transform block (TB) size. The maximum transform block size allowed for HEVC is 32x32 pixels, instead of 16x16 pixels in H264/AVC standard. Thus, to get more accurate predictions when the size of the current transform block is bigger than 16 x 16 pixels, HEVC use 33 different directional orientations for angular prediction. And for different transform blocks, the maximum number of intra-prediction modes available is different. In HEVC, along with the DC and planar mode, there are 35 intra prediction modes, as illustrated in Fig. 2.

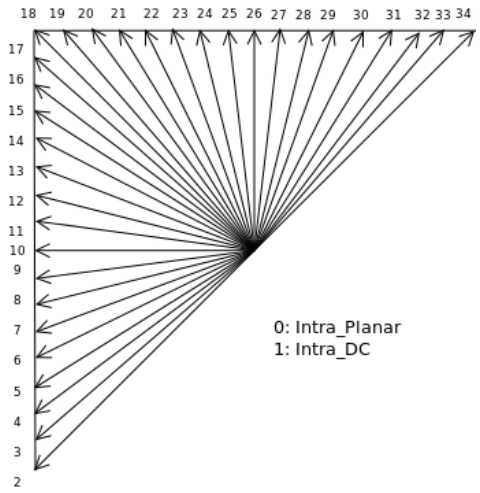

Figure 1. 35 Intra Prediction Modes In Hevc

2.4. Improved Motion Information Encoding

In H.264/AVC standard, the motion information is transmitted in two ways according to whether the skip mode is used or not. In the skip mode, no motion information is transmitted to the decoder, and the current block uses the motion information from the collocated block in the predefined reference picture. In non-skip mode, the difference of the predicted motion vector and the actual motion vector is transmitted to the decoder. The predicted motion vector is the median of three surrounding motion vectors. HEVC derives the motion information both spatially and temporally. In addition, HEVC introduces the merge mode. In merge mode, the decoder needs to create a candidate list, which contains the motion information from neighboring prediction unit (PU), either spatially or temporally, and then the decoder uses the element, specified by the index, transmitted by the encoder, as the motion information of the current prediction unit. In non-merge mode, the normal motion information, reference picture index and motion vector difference, are transmitted to the decoder. The decoder also needs to create a candidate list, which contains the motion information from neighboring blocks. The length of the candidate list is limited to two to reduce the complexity of the HEVC standard, especially for the encoder.

2.5. Improved Non-Integer Pixel Interpolation

As in H.264/AVC standard, HEVC supports motion vectors with units of one quarter of the distance between luma samples, and one eighth of the distance between chroma samples for 4:2:0 sampling. However, there are some improvements in both the precision of the interpolation and the complexity when implementing the interpolation process. In the interpolation process for luma samples, an 8-tap filter is used for half-sample positions, and a 7-tap filter is used for quarter-sample positions. Compared with the 6-tap filter used in H.264/AVC standard, the interpolation process in HEVC can get better precision because of longer filters. On the other hand, HEVC uses separable application of filters for half-sample and quarter-sample positions, while H.264/AVC applies a two-stage interpolation process. Obviously, without the intermediate operations, the architecture of the interpolation process in HEVC can be simplified, with less complexity.

2.6. Better Approximated Transform Coefficients

To transform the residual coefficients, the HEVC standard applies a DCT-like integer transform, just as what's shown in the previous H.264/AVC standard. The transform is done according to the size of the transform block (TB), which ranges from 4 x 4 pixels to

32 x 32 pixels. The integer transforms used in HEVC are better approximations of the DCT than the transforms used in H.264/AVC. Besides, HEVC can use an alternative 4x4 discrete sine transform(DST) for 4 x 4 transform blocks, because the DST basis functions start low and increase, compared with the DCT basis functions that start high and decrease [11]. Thus, the DST transform is supposed to be more suitable for blocks coded with some directional intra-prediction modes.

2.7. Improved Entropy Coding

In H.264/AVC standard, two entropy coding methods, CAVLC and CABAC are proposed. While in HEVC, only the CABAC entropy coding is used. The arithmetic coding engine of the CABAC remains the same as what's in H.264/AVC standard, but there is some optimization to reduce the memory usage and to improve the parallelization. First, great effort is made to reduce the number of contexts associated with the residual syntax when in context modeling stage. This reduction contributes to lower the amount of memory required by the entropy decoder and the cost of initialing the engine. Besides, parallel context processing is enabled, thus the decoder can derive multiple context indices in parallel, which improves the throughput of the entropy decoding process. Other detailed improvements of the entropy coding in HEVC can be found in Ref. [12].

2.8. Alternative Sao Filter

HEVC introduces a new filter, which is called sample adaptive offset (SAO) filter, after the normal de-blocking filter in the loop-filter process. The SAO is a nonlinear filter, which allows additional refinement of the reconstructed samples, and it enhances the sample representation in both smooth areas and around edges. It modifies the decoded samples by conditionally adding an offset value to each sample after the application of the de-blocking filter, based on values in look-up tables transmitted by the encoder. As an additional stage with respect to the loop-filter in H.264/AVC, increased complexity is introduced by SAO in the HEVC standard.

3. Analysis of HEVC Architecture

3.1. HEVC Encoder

Similar to the encoder of H.264/AVC, the HEVC encoder contains lots of different coding components. The encoder reads the raw video sequence, and then encodes it into all kinds of syntax elements encapsulated in the final bit-stream. A general architecture of the HEVC encoder could be illustrated with Figure 3.

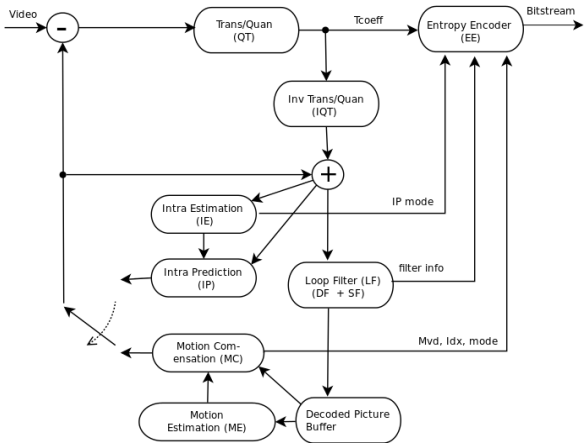

Figure 1. General Architecture Of The HEVC Encoder

As can be shown, in the HEVC encoder, the video sequence is encoded as the following syntax elements, the coefficients of the residual block (Tcoeff), the intra prediction info (IP mode), the motion information (Mvd, Idx, mode information), filter control information (filter info) and some associated control signals. The rest of this part gives a brief introduction of each component of the encoder.

Motion estimation (ME): This part aims to find the best prediction if inter prediction is selected. Unlike the H.264/AVC encoder, HEVC encoder searches the best prediction not only from the spatial neighbors, but also from the temporal neighbors. The output of this stage is the motion information, including the difference of the motion vector (Mvd), the index of the reference picture (Idx), and the inter mode information. In a normal encoding process, this part is usually an extremely time-consuming part, since large amount of computation is needed in this stage.

Motion compensation (MC): This component uses the output of the motion estimation to get the prediction for the current picture. The output of this stage is to be subtracted from the current picture to get the residual picture, which is served to the transform and quantization process.

Intra estimation (IE): In intra prediction stage, the HEVC encoder goes over the 35 intra prediction modes to find out which one is the best. This process is very time-consuming as well. The output of this stage is the intra mode information needed for the decoder.

Intra prediction (IP): This part uses the intra prediction mode information to get the prediction for the current block or the current picture. Before the data is served into the transform/quantization (QT) component, the output of this stage is subtracted from the current block or current picture.

Transform and quantization (QT): The sample values are transformed from the temporal domain into the frequency domain, and then most of the coefficients are removed during the quantization stage. In most situations, the quantized coefficients of the residual picture are transmitted to the decoder. There are two kinds of transforms In HEVC, the discrete cosine transform (DCT) and the discrete sine transform (DST). The output of this stage is the quantized coefficients of the residual picture (Tcoeff).

Inverse transform and inverse quantization (IQT): In the encoder, the coded picture or block is reconstructed so that they could be used in the following intra or inter estimation process. Thus, the quantized coefficients of the residual block are processed in this stage to get the original residual temporal sample values.

Loop filter (LF): The HEVC encoder uses a two-stage loop filter for the reconstructed picture. First, a deblocking filter (DF) is used to remove the blocking artifacts. Then a SAO filter (SF) is performed to refine the sample values. In this stage, the necessary filter control information is transmitted to the decoder.

Entropy encoder (EE): The HEVC encoder only implements the CABAC entropy encoder. As shown in the architecture, the entropy encoder encodes the coefficients (Tcoeff), the motion information (Mvd, Idx, mode), the intra prediction information (IP mode), filter control data (filter info), along with some global control signals. Output of this component is the bit-stream constrained with the HEVC video coding standard.

3.2. HEVC Decoder

Input to the HEVC decoder is the bit-stream constrained with the HEVC standard. The decoder reads the bit-stream, extracts and decodes all the syntax elements, and then constructs each frame of the original video sequence. A common architecture of the HEVC decoder could be illustrated with Figure 4.

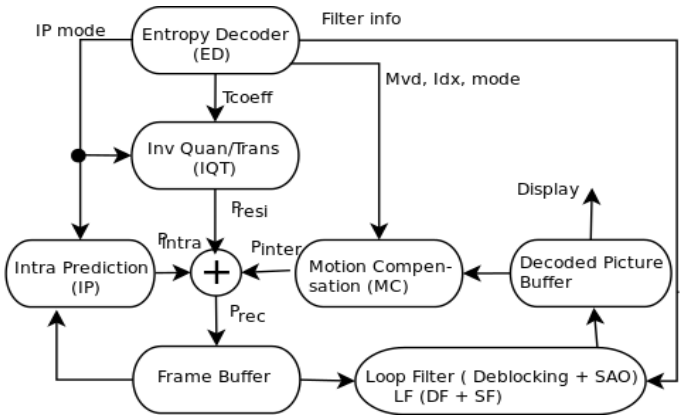

Figure 1. General Architecture Of The HEVC Decoder

As can be seen from above, the HEVC decoder consists of five main components, the entropy decoder (ED), the intra prediction (IP) part, the motion compensation (MC) part, the inverse transform and inverse quantization (QT) part, and the loop filter (LF) part. The rest of this section gives a brief introduction of each component.

Entropy decoder (ED): The entropy decoder in HEVC only implements the CABAC decoder. It reads the bit-stream, decodes all the syntax elements, and serves the decoded information for the other components of the decoder. The decoded elements include the motion information (the motion vector residual signals Mvd, the referenced picture index Idx, other control signals), the intra-prediction mode information (IPmode), the transform coefficients (TCOEFFs) and the necessary global control signals.

Inverse quantization and inverse transform (IQT): This part first applies the inverse quantization process to get the scaled coefficients. Then proper inverse transform is performed according the transform block size. To reduce the computation, the 2-D transform is separated into two 1-D transforms. For HEVC decoder, this part needs to incorporate both the inverse DCT and DST transform. The output of this part is marked as P_{resi}.

Intra prediction (IP): This part uses the intra prediction mode information (IP mode) to derive the prediction block for the current block coded in intra-prediction mode. Then the decoded picture P_{intra} is added to the decoded residual picture P_{resi} to reconstruct the current picture P_{rec}.The decoded intra-predicted picture is stored into to the frame buffer, serving as the reference picture when needed.

Motion compensation (MC): This part is intended to decode inter-predicted pictures. First, the motion information should be derived so that the decoder could know where to get the prediction for current picture. Before the prediction is derived, a non-integer sample interpolation is performed when the motion vector is not in units of integer. Then the derived prediction P_{inter} is added to the residual picture P_{resi} to reconstruct the current picture P_{rec}. The decoded inter-predicted picture is also stored into the decoded picture buffer (DPB) to serve as the reference picture when needed.

Loop filter (LF): First, a normal de-blocking filter (DF) is applied on the edges of the boundaries of coding blocks, prediction blocks, and transform blocks to remove

the artifacts caused by the block-based processing. Then a SAO filter (SF) process is applied by conditionally adding an offset value to each sample.

4. Evalution of HEVC Reference Software

4.1. Test Conditions and Test Sequences

The experiment profiles the HM12.0, and the experimental environment is described as shown in Table 1.

Table 1. Test Environment

Processor	Intel Core i7-4770 (3.4GHz x 4)
Memory	8GB
Operating System	64-bit Linux (Kernel: 3.8.0-29-generic)
Compiler	g++-4.6.3

The test sequences recommended by JCT-VC for the experiment are listed in Table 2. There is a set of encoder configurations, used when generating the bit-streams for this paper, which is given by the JCT-VC community [13], including:

1) all intra (AI), only I slices exist in the bitstream;
2) Random access (RA), where picture reordering is used in a pyramidal structure with a random-access picture about every 1 second. This configuration emulates what may be used in a broadcasting environment.
3) Low delay (LD), where no picture reordering is used. And only the first frame is encoded as an I-slice. This configuration emulates what may be used in a video conferencing environment.

The bit-streams used in this paper:

(1) cover different encoder configurations (AI, RA, LD), when generated using HM encoder.
(2) cover different quantization parameters (QP) (22, 27, 32, 37), when generating using HM encoder.
(3) cover different video content with resolutions ranging from 240p (416 x 240 pixels) to 1600p (2560 x 1600 pixels).

Table 2. Test Sequences

Class	Sequence	# of frames	Bit depth (bpp)	Frame rate (fps)
A(1600p)	Traffic	150	8	30
	PeopleOnStreet	150	8	30
B(1080p)	Kimono	240	8	24
	ParkScene	240	8	24
	Catus	500	8	50
	BQTerrace	600	8	60
	BasketballDrive	500	8	50
C(480p)	BQMall	600	8	60
D(240p)	RaceHorses	300	8	30
	BQSquare	600	8	60
E(720p)	Vidyo1	600	8	60
	Vidyo3	600	8	60
	Vidyo4	600	8	60

4.2. Profiling Results for HM Encoder

This part mainly focuses on the performance of the HEVC standard. In this part, the HM12.0 encoder for HEVC standard is evaluated with random-access configuration, which is the most common situation in reality. Besides, the JM18.5, which is the encoder for H.264/AVC high profile, is also analyzed with corresponding configurations. For the evaluation of the encoders, PSNR is used to measure the quality of the compression. Table 3 shows the profiling results.

Table 3. Profiling Results for Encoder

Sequence	Bit-rate (kbps)		PSNR (db)	
	HM12.0	JM18.5	HM12.0	JM18.5
BasketballDrive	1521.4	3533.0	38.0	38.37
BQTerrace	1296.9	2120.1	37.3	37.7
Catus	1489.9	2783.6	36.5	37.2
Kimono1	534.3	1239.7	38.8	39.4
ParkScene	671.5	1308.5	36.4	37.0
BQMall	501.2	926.5	36.8	37.6
BQSquare	183.9	268.5	35.7	36.4

As can be seen, the HEVC standard outperforms the H.264/AVC standard, with the bit-rate saving ranging from 35.7% to 56.9%. Fig. 5 illustrates the average bit-rate saving of video contents varying from 240p to 1080p. It's shown that HEVC is more preferable for video content with higher resolutions, in which a doubling of the coding efficiency can be easily achieved, delivering nearly the same visual quality as the previous H.264/AVC standard at the same time.

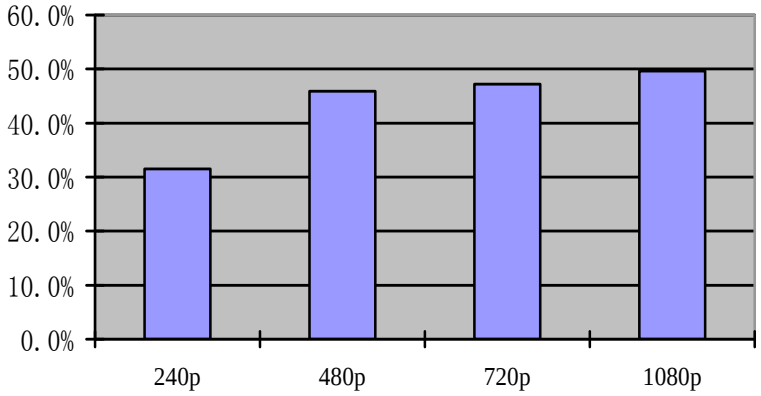

Figure 5. General Architecture Of HEVC Decoder

4.3. Profiling Results for HM Decoder

Table 4 shows the decoding time for the test sequences listed above with different encoder configurations.

It is shown that the decoding time for bit-streams encoded with all-intra configuration is much bigger than the other two types due to the existing of large amount of temporal redundancy when coded in all-intra mode. While the difference of decoding time between random-access and low-delay is quite small. And at this point, it's natural to point out the HM decoder, without certain optimization in parallelization, is not sufficient to decode videos with resolutions equal to or higher than 1080p for real-time applications.

To further analyze the HM decoder, the HM decoder is profiled to find out the complexity of each component of the decoder under all-intra and random-access configuration, with the results shown in Table 5 and Table 6 separately.

Table 4. Decoding Time with Different Configurations

Class	Sequences	# of frames	Total decoding time(s)		
			AI	RA	LD
A(1600p)	Traffic	150	42.3	12.3	12.7
	PeopleOnStreet	150	48.3	20.6	20.2
B(1080p)	Kimono	240	22.9	11.5	11.6
	ParkScene	240	35.6	11.6	12.0
	Catus	500	69.6	21.2	22.8
	BQTerrace	600	93.0	26.2	26.8
	BasketballDrive	500	57.3	26.3	25.7
C(480p)	BQMall	600	19.3	5.7	5.8
D(240p)	RaceHorses	300	2.7	1.1	1.1
	BQSquare	600	6.5	1.5	1.5
E(720p)	Vidyo1	600	30.7	8.2	9.4
	Vidyo3	600	54.3	8.3	14.4
	Vidyo4	600	50.1	12.4	12.6

Table 5. Share of the Decoding Time with AI Configuration (QP=32)

Sequence	ED (%)	MC (%)	IQT (%)	PR (%)	DF (%)	SF (%)
Traffic	8.1	43.6	2.8	4.4	15.7	3.5
PeopleOnStreet	13.5	24.4	4.3	7.6	19.5	2.8
ParkScene	8.7	47.4	2.9	4.7	14.1	2.9
Catus	9.2	38.1	5.0	6.9	15.9	3.2
BQTerrace	7.8	48.4	2.6	4.1	13.0	3.3
BasketballDrive	7.8	42.0	6.2	7.6	14.5	2.7
BQMall	11.4	40.5	4.1	7.4	13.9	2.6
RaceHorses	16.1	36.3	5.1	9.6	14.6	2.3
BQSquare	13.7	46.8	2.9	5.9	11.3	1.9
Vidyo1	6.2	41.2	3.4	4.4	13.8	4.2
Vidyo3	7.0	44.2	4.1	5.0	15.1	3.7
Vidyo4	6.4	42.7	4.0	4.8	15.1	4.1
Average	9.4	42.6	4.0	5.9	14.7	3.1

Table 6. Share of the Decoding Time with AI Configuration (QP=32)

Sequence	ED (%)	MC (%)	IQT (%)	PR (%)	DF (%)	SF (%)
Traffic	12.4	-	20.6	29.4	13.5	2.9
PeopleOnStreet	11.8	-	18.7	31.0	13.2	2.4
ParkScene	13.5	-	20.6	29.1	12.7	2.8
Catus	12.1	-	21.2	29.6	12.8	2.6
BQTerrace	13.5	-	19.8	30.2	11.9	2.4
BasketballDrive	9.7	-	24.8	28.7	13.6	3.1
BQMall	13.7	-	18.8	31.5	10.9	2.4
RaceHorses	16.9	-	17.5	31.1	9.8	2.6
BQSquare	19.6	-	16.1	31.4	8.5	1.5
Vidyo1	8.8	-	24.5	29.5	13.2	3.1
Vidyo3	8.5	-	27.3	30.6	12.5	2.8
Vidyo4	8.5	-	27.1	30.2	12.8	3.4
Average	12.2	-	22.1	29.8	12.3	2.7

In these two tables, the component entropy decoder, motion compensation, intra-prediction and intra reconstruction, deblocking filter, SAO filter are abbreviated as ED, MC, IQT, PR, DF and SF. For each sequence, the total share of the six components is not 100%, since the HM decoder need to do some other stuffs, including reading the bit-stream, collocate buffers for different data structures, writing the reconstructed YUV, and so on. These stuffs would be marked as the MISC part. The share of each component in AI or RA configurations is depicted in Fig. 6. In all-intra mode, the average complexity of ED, MC, IQT, PR, DF, MISC and SF is 12.2%, 0%, 22.1%, 29.8%, 12.3%, 21.90%, 2.7%. The intra-prediction and inverse quantization/transform process take nearly half of the decoding process. In random-access mode, the component ED, MC, IQT, PR, DF, MISC, SF shares 9.4%, 42.6%, 4.0%, 5.9%, 14.7%, 20.40% and 3.1% of the complexity of the HM decoder. The motion compensation part dominates the decoding process in random-access mode, taking nearly the half of the whole time. Thus, special attention should be paid to the motion compensation part when designing a real-time HEVC

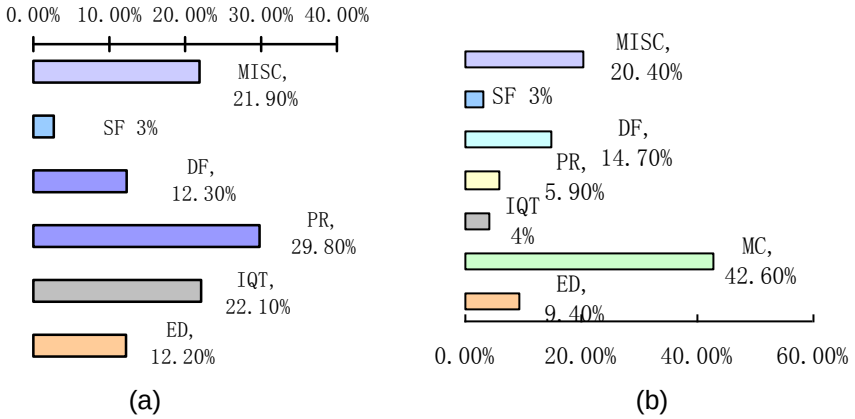

Figure 6. The Share Of Each Component In The Decoder (A) All-Intra Configuration (B) Random-Access Configuration

decoder.

At the end of the evaluation, further analysis is done on the influences of the quantization parameters (QP). The profiling results are shown in Table 7 and Table 8. As can be seen from above, the quantization parameters not only affect the total decoding time, but also affect the time taken by each component of the decoder. As the QP increase from 22 to 32, the decoding time decreases, however, if the QP continues to grow up to 37, the decoding time may increase. For each component of the decoder, when the QP grows from 22 to 37, the shares of QT, PR, DF become bigger and bigger, while the share of ED decreases because more coefficients is removed in the quantization process with larger QP. For random-access configuration, the share of each component with different quantization parameters (QP) is depicted in Fig. 7.

Table 7. Decoding Time (S) With Diifferent QP

Sequence	AI				RA			
	22	27	32	37	22	27	32	37
Traffic	60	50	42	36	31	24	12	17
PeopleOnStreet	101	85	48	63	52	40	20	26
Kimono	29	26	22	20	15	13	11	10
ParkScene	52	43	35	28	17	14	12	10
Catus	176	124	69	99	63	40	22	29
BQTerrace	246	172	93	127	99	51	26	38
BasketballDrive	89	68	57	49	42	30	25	23
BQMall	47	40	19	27	14	10	5	8
RaceHorses	6	5	2	4	3	2	1	2
BQSquare	8	7	6	5	2	2	2	1
Vidyo1	40	34	30	26	12	10	9	9
Vidyo3	65	55	54	47	19	16	14	13
Vidyo4	70	51	50	42	17	15	12	12

Table 8. Share of Each Component with Different Configurations and QP

CfgType	QP	ED(%)	MC(%)	IQT(%)	PR(%)	DF(%)	SF(%)
AI	22	21.7	-	17.8	28.5	9.5	2.2
	27	15.8	-	20.3	29.6	11.1	2.8
	32	12.2	-	22.1	29.8	12.3	2.7
	37	8.8	-	25.5	30.2	13.2	2.5
RA	22	18.1	39.5	3.6	6.2	13.7	3.7
	27	12.6	42.3	3.9	6.2	14.5	3.2
	32	9.4	42.6	4.0	5.9	14.7	3.1
	37	7.4	45.2	4.3	5.5	14.9	3.0
LD	22	21.6	41.8	1.4	2.7	14.7	3.9
	27	14.8	43.9	1.4	2.5	16.5	3.7
	32	11.3	42.9	1.5	2.3	17.6	3.4
	37	9.1	42.9	1.7	2.2	18.3	3.2

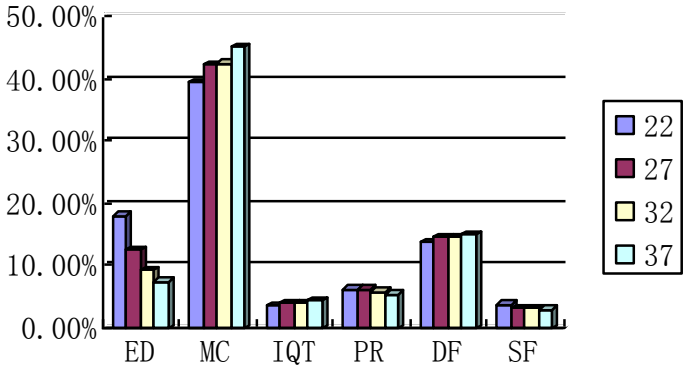

Figure 7. The Share Of Each Component In The Decoder With Different QP Under Random-Access Configuration

5. Conclusion

This paper highlights the new features of the HEVC video coding standard, and then evaluates the complexity of the standard by proofing the HEVC reference software. The HEVC standard can get twice the coding efficiency, delivering the same or better visual quality at the same time. In the HEVC decoder, the motion compensation part takes nearly half the decoding time in the normal decoding process. Besides, the loop filter and the intra-picture decoding part also take a large percent of the decoding time in most cases. Different quantization parameters have a great influence not only on the whole decoding time, but also on each component's share of the decoder. Besides, for videos with resolutions no less than 1080p, the reference software cannot serve as a real-time decoder without special optimization.

Acknowledgements

This work is supported by Natural Science Foundation of Shandong Province ZR2013FQ006, Fund of independent innovation in Shandong Province 2013CXB3020, and State Key Laboratory of digital multimedia technology 2013-1-2569, Shandong post-doctor innovation foundation grant 201002029. The authors would like to thank all research partners for the significant contributions in this work.

References

[1] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)" IEEE trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1669-1684, Dec. **2012**.
[2] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "The First Edition of High Efficiency Video Coding (HEVC) Text Specification", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Apr, **2013**.
[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1648–1667, Dec. **2012**.
[4] S. Ma and C.-C. J. Kuo, "High-definition video coding with super-macroblocks," in Proc. Visual Commun. Image Process, vol. 6508. Jan. 2007, p. 650816.
[5] P. Chen, Y. Ye, and M. Karczewicz, "Video Coding Using Extended Block Sizes", VCEG document AJ23, 36th VCEG Meeting, Oct. **2008**.
[6] T. Yoshino, S. Naito, and S. Sakazawa, "Preliminary Response for Draft Call for Evidence on High Performance Video Coding", MPEG document M16082, 87th MPEG Meeting, Feb. **2009**.

[7] S. Sekiguchi and S. Yamagishi, "On Coding Efficiency With Extended Block Size for UHDTV", MPEG document M16019, 87th MPEG Meeting, Feb. **2009**.
[8] Il-Koo Kim , Junghye Min , T.Lee, Woo-Jin Han ,JeongHoon Park , "Block Partitioning Structure in the HEVC Standard", IEEE Transactions on Circuits and Systems for Video Technology, Volume:22,Issue:12, pp: 1697-1706.
[9] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis", IEEE trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1685-1696, Dec. **2012**
[10] K. McCann, W.-J. Han, I.-K. Kim, J. H. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M. S. Cheon, and N. Shlyakhov, Samsung's Response to the Call for Proposals on Video Compression Technology, JCT-VC document A124, 1st JCT-VC Meeting, Apr. **2010**
[11] J. Han, A. Saxena, and K. Rose, "Towards jointly optimal spatial prediction and adaptive transform in video/image coding," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP),Mar. **2010**, pp. 726–729.
[12] V. Sze, and M. Budagavi, "High Throughout CABAC Entropy Coding in HEVC", IEEE trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp: 1778-1791, Dec, **2012**.
[13] F. Bossen, "Common test conditions and software reference configura-tions," document JCTVC-H1100 of JCT-VC, San Jos´ CA, USA, Feb. **2012**.

Authors

Jianjun Wang was born in September, 1989, in China. He received the BS. Degree in Integrated Circuit Design and Integrated System from Shandong University, China, in 2011, and now he is pursing the MS. Degree in Integrated Circuit Design in Shandong University. His current research interests include VLSI design, SoC design, and video coding design.

Li Zhou received Ph.D degree on 2004 from Zhejiang University, China. In 2004, Dr.Zhou joined Freescale semiconductor R\&D center as principle design architecturer till 2009. As an architecturer and core team member, Dr. Zhou participated in multiple National high-tech SoC and HDTV fundamental research projects, led multiple 65nm/90nm 10 million gate scale VLSI SoC chips, and joined many VLSI project and architecture researches. From 2009, Dr. Zhou is a assistant professor in Shandong University, China. Her current research interests include stereo vision system algorithm and hardware design, GPU architecture and VLSI design, high performance processor architecture and VLSI design, etc.

Tao Sun received Ph.D degree on 2002 from Zhejiang University, China. Dr. Sun joined Suzhou China Core Co. Ltd as a researcher on CPU architecture and vice-general manager till 2007. From 2007 to 2009, Dr. Sun was a senior manager in Spansion China, led VLSI design projects. From 2009 till now, Dr. Sun is an associated professor in University of Jinan, China. His research interests include CPU/VLSI architecture, Solid storage architecture, etc.

