

Complexity Analysis of Next-Generation HEVC Decoder

Marko Viitanen, Jarno Vanne, Timo D. Hämäläinen,
Moncef Gabbouj
Tampere University of Technology
Tampere, Finland

Jani Lainema
Nokia Research Center
Nokia Corp.
Tampere, Finland

Abstract— This paper analyzes the complexity of the HEVC video decoder being developed by the JCT-VC community. The HEVC reference decoder HM 3.1 is profiled with Intel VTune on Intel Core 2 Duo processor. The analysis covers both Low Complexity (LC) and High Efficiency (HE) settings for resolutions varying from WQVGA (416×240 pixels) up to 1600p (2560×1600 pixels). The yielded cycle-accurate results are compared with the respective results of H.264/AVC Baseline Profile (BP) and High Profile (HiP) reference decoders. HEVC offers significant improvement in compression efficiency over H.264/AVC: the average BD-rate saving of LC is around 51% over BP whereas the BD-rate gain of HE is around 45% over HiP. However, the average decoding complexities of LC and HE are increased by 61% and 87% over BP and HiP, respectively. In LC, the most complex functions are motion compensation (MC) and loop filtering (LF) that account on average for 50% and 14% of the decoder complexity. The decoding complexity of HE configuration is on average 42% higher than that of the LC configuration. Majority of the difference is caused by extra LF stages. In HE, the complexities of MC and LF are 37% and 32%, respectively. In practice, a standard 3 GHz dual core processor is expected to be able to decode 1080p HEVC content in real-time.

Index Terms — High efficiency video coding (HEVC), HEVC Test Model (HM), video decoding, complexity analysis.

I. INTRODUCTION

The wireless and wired transmission of next-generation resolutions demand coding efficiency that is beyond the capabilities of the current state-of-the-art H.264/AVC standard [1]. Therefore, MPEG and VCEG have established a Joint Collaborative Team on Video Coding (JCT-VC) to develop a successor to H.264/AVC. This forthcoming international standard is referred to as *High Efficiency Video Coding (HEVC)* [2]. HEVC focuses on coding of progressively scanned rectangular pictures whose resolution can vary at least between QVGA (320×240) and UHDTV (7620×4320).

The plan of JCT-VC is to publish draft versions of HEVC in 2012 and the final standard in early 2013. JCT-VC is currently in a collaborative phase refining the technical content of the draft design that has originally been created from the best-performing initial HEVC proposals [3]-[7]. The initial HEVC versions roughly halve the bit rate over H.264/AVC with the same subjective visual quality, whereas the respective BD-rate savings have been measured to be around 20 - 40% [8]. To be able to study trade-offs between complexity and coding

efficiency, the HEVC coding tools are separately specified for *Low Complexity (LC)* and *High Efficiency (HE)* operation [9].

The public HEVC assessments have mainly focused on its BD-rate and BD-PSNR gains [3]-[8], whereas the complexity evaluation of HEVC is limited either to single HEVC tools [10] or processing time comparisons between consecutive HEVC versions and H.264/AVC [8], [11]. This paper addresses the cycle-accurate complexity of the HEVC reference decoder and compares the results with H.264/AVC reference decoder.

Since the standardization is still in progress, the experiments rely on the temporary *HEVC Test Model* HM 3.1 utilizing both LC and HE *random access (RA)* configurations [9]. HM 3.1 is benchmarked against the current JM 18.0 [12] reference decoder of H.264/AVC. The used JM profiles are *Baseline Profile (BP)* and *High Profile (HiP)*. All cycle-accurate profiling results are yielded with Intel® VTune™ Amplifier XE 2011 on Intel® Core™2 Duo E8400 processor.

The remainder of this paper is organized as follows. Section II presents the HEVC decoder and its main functions. Section III describes the setup for the complexity analysis and reports the cycle-accurate complexities of HM 3.1 LC and HE decoders. Section IV compares the complexities of HM 3.1 and JM 18.0. In addition, practical implementation alternatives for the HEVC decoders are discussed. Section V concludes the paper.

II. HEVC DECODER

The coding structure of HEVC is based on a quadtree scheme in which the size of the square-shaped *Coding Unit (CU)* is $2N \times 2N$, where $N \in \{4, 8, 16, 32\}$. Each CU can be recursively divided into four smaller CUs until $N = 4$. In inter/intra prediction, the CUs can be further partitioned into rectangular-shaped *Prediction Units (PUs)*. With CU of size $2N$, the size of the PU can be $2N \times 2N$, $2N \times N$, $N \times 2N$, or $N \times N$ [13]. For transforms, HEVC specifies *Transform Unit (TU)* whose size can be from 4×4 to 32×32 .

A general HEVC decoder structure and its main functions are depicted in Fig. 1. *Entropy decoder (ED)* stage extracts binary codewords from the input bitstream and converts them to original syntax elements including *intra prediction mode (IP mode)*, *quantized transforms coefficients (TCOEFFs)*, *motion vectors (MVs)*, and *indexes (Idxs)* into a reference picture list. HEVC uses two ED algorithms: *content-adaptive variable length coding (CAVLC)* with LC settings and *content-adaptive*

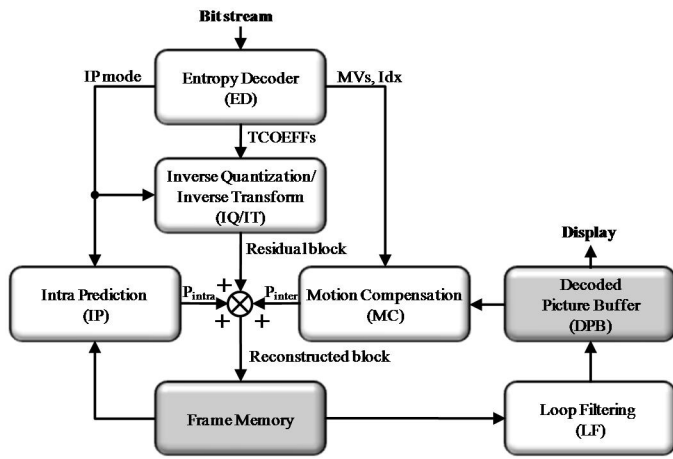


Figure 1. HEVC decoder model

binary arithmetic coding (CABAC) with HE settings. Their basic operating principles are inherited from H.264/AVC.

Inverse quantization and inverse transform (IQ/IT) stage dequantizes and transforms frequency-domain TCOEFFs back to spatial domain *residual blocks*. HEVC uses integer *Discrete Cosine/Sine Transform (DCT/DST)* for which all transform matrices have been upgraded from H.264/AVC with added precision in the integer scale.

Intra prediction (IP) stage accesses a *frame memory* to compute *intra prediction* (P_{intra}) for a decoded block. The frame memory contains previously decoded blocks of the current picture. HEVC increases angular IP modes over H.264/AVC by specifying 17 modes for 4×4 blocks, 34 modes for 8×8 , 16×16 , and 32×32 blocks, as well as 3 modes for 64×64 blocks. In addition, HEVC contains planar IP mode. If the decoder operates in IP mode, P_{intra} is added to a residual block and a *reconstructed block* is stored in the frame memory.

Motion compensation (MC) stage produces an *inter prediction* (P_{inter}) for a decoded block by addressing *decoded picture buffer (DPB)* with MVs and Idxs. DPB contains previously decoded pictures. HEVC uses 8-tap interpolation filter for luminance and 4-tap filter for chrominance samples in $\frac{1}{8}$ -pixel (chrominance only), $\frac{1}{4}$ -pixel, and $\frac{1}{2}$ -pixel MC. If the decoder operates in inter prediction mode, P_{inter} is added to the residual block to form the reconstructed block.

Loop filtering (LF) stage filters the distortions and visible CU/PU/TU borders from the picture. LF stage contains three in-loop filters: *deblocking filter (DF)*, *adaptive loop filter (ALF)*, and *sample-adaptive offset (SAO)*. DF corresponds to DF in H.264/AVC, ALF improves quality with diamond-shape 2D filters [13], and SAO applies offset values indicated in the bitstream [13]. Each filter can be used sequentially according to encoder decision. ALF is excluded from LC.

III. HEVC DECODER ANALYSIS

TABLE I tabulates the test environment. SIMD extensions of the processor have not been exploited in order to maintain platform-independency. The analysis relies on VTune which is able to provide cycle counts for each function of HM. During

the analysis, HM was the only software running to reduce noise caused by other computer processes on the results.

The analysis is based on HM 3.1 with two settings (RA-LC and RA-HE) and four *quantization parameter (QP)* values (22, 27, 32, and 37). TABLE II lists the test sequences recommended by JCT-VC for the selected HM settings. The sequences have been encoded according to the JCT-VC Common Test Conditions for RA settings [9] with *I*-frames roughly at one second intervals and limiting the number of reference pictures in inter prediction to four. Each HM configuration has been run 10 times with all sequences and median of the sequence-specific test runs have been selected.

The profiling results are tabulated in TABLE III and TABLE IV, in which only the sequences with maximum and minimum complexities are reported for each format. The absolute complexities of the tabulated sequences are reported as *million cycles per frame (Mcpf)*. In addition, the percentages of the cycle counts are allocated for each decoder stage (ED, IQ/IT, IP, MC, and LF). Pre-processing, memory, and post-processing functions not belonging directly to any of these stages are allocated to group “Misc”.

In LC, the average complexities of ED, IQ/IT, IP, MC, and LF are 5%, 6%, 2%, 50%, and 14%, respectively. Changing settings from LC to HE increases complexity of HM by 42% of which the majority is caused by ALF overhead in HE. The respective function-specific shares of HE are 6%, 5%, 1%, 37%, and 32%. When QP is changed from 22 to 37 in LC, the complexities of ED, IQ/IT, IP, MC, and LF degrade 80%, 57%, 67%, 23%, and 34%, respectively. In HE, the respective degradations are 88%, 66%, 61%, 23%, and 54%.

TABLE I. TEST ENVIRONMENT

Processor	Intel Core 2 Duo E8400 (2 × 3.0 GHz)
Memory	8 GB
L1 Cache	2 × 32 KB (instruction) + 2 × 32 KB (data)
L2 Cache	6 MB
Compiler	Microsoft Visual C++ 2010
Operating system	64-bit Microsoft Windows 7 Enterprise SP 1

TABLE II. TEST SEQUENCES

Format	Sequence	# of frames	Bit depth (bpp)	Frame rate (fps)
2560×1600 (1600p)	Traffic	150	8	30
	PeopleOnStreet	150	8	30
	Nebuta	300	10	60
	SteamLocomotive	300	10	60
1920×1080 (1080p)	Kimono	240	8	24
	ParkScene	240	8	24
	Cactus	500	8	50
	BQTerrace	600	8	60
	BasketballDrive	500	8	50
832×480 (480p)	RaceHorses	300	8	30
	BQMall	600	8	60
	PartyScene	500	8	50
	BasketballDrill	500	8	50
416×240 (240p)	RaceHorses	300	8	30
	BQSquare	600	8	60
	BlowingBubbles	500	8	50
	BasketballPass	500	8	50

TABLE III. THE WORST-CASE TEST SEQUENCES

Seq.	Prof.	QP	ED (%)	IQ/IT (%)	IP (%)	MC (%)	LF (%)	Misc (%)	Total (Mcpf)
Nebuta (1600p)	HE	22	18.5	14.0	0.6	17.1	28.0	21.7	2306.2
		27	12.0	14.3	0.5	23.4	34.7	15.2	1939.3
		32	5.9	8.5	0.2	30.0	43.0	12.5	1436.2
		37	2.4	3.5	0.2	34.1	45.5	14.3	975.5
	LC	22	10.0	19.9	1.6	25.8	6.4	36.3	1358.8
		27	8.3	18.7	1.3	36.8	10.3	24.7	1179.2
		32	4.8	12.9	0.5	51.0	13.9	17.0	809.4
		37	2.5	5.2	0.3	59.6	15.7	16.7	585.3
BasketballDrive (1080p)	HE	22	8.3	6.1	1.6	27.4	39.8	16.8	736.5
		27	4.2	5.5	1.2	33.1	39.8	16.2	561.2
		32	2.3	4.6	0.9	34.6	42.4	15.1	512.0
		37	1.5	4.0	0.7	37.7	41.4	14.6	459.2
	LC	22	6.0	6.8	3.1	44.8	15.0	24.3	460.8
		27	4.0	7.3	2.2	50.7	15.3	20.4	370.7
		32	2.9	6.9	1.6	54.6	15.2	18.7	327.4
		37	2.1	6.0	1.2	56.8	16.2	17.6	306.5
RaceHorses (480p)	HE	22	12.8	5.0	1.8	25.0	35.0	20.4	188.6
		27	7.8	4.7	1.7	28.8	37.8	19.1	139.9
		32	5.0	4.4	1.6	32.8	37.8	18.3	109.6
		37	3.5	4.6	1.4	39.0	32.8	18.6	84.3
	LC	22	9.0	7.0	3.2	37.8	13.2	29.7	121.9
		27	6.5	6.9	2.9	42.6	15.2	25.8	93.4
		32	4.5	6.7	2.5	46.0	16.9	23.4	77.4
		37	3.2	6.6	1.9	49.5	17.0	21.7	67.1
RaceHorses (240p)	HE	22	13.3	3.7	1.9	30.2	27.7	23.2	44.5
		27	9.4	3.4	1.9	33.8	29.9	21.6	34.2
		32	6.3	3.3	1.9	39.6	27.5	21.3	25.7
		37	4.0	3.3	1.4	46.1	23.6	21.6	20.2
	LC	22	9.1	4.7	2.9	41.3	12.0	30.1	32.0
		27	7.0	4.6	2.8	45.2	13.0	27.4	25.2
		32	4.8	4.7	2.6	48.1	14.5	25.3	21.0
		37	3.7	5.2	1.8	52.0	13.8	23.5	17.4

IV. COMPARISON OF HEVC AND H.264/AVC DECODERS

Fig. 2 depicts average QP-specific complexities of HM LC/HE and JM BP/HiP at each resolution. As in LC/HE, hierarchical coding structure is also used in BP/HiP. Since 10-bit precision is not supported by BP/HiP, only 8-bit sequences are compared. On average, LC is 61% more complex than BP and the respective ratio is 87% between HE and HiP. LC reduces complexity of ED by 20% over BP, whereas the LC overheads of IQ/IT, IP, MC, and LF are 2.3x, 2.3x, 2.2x, and 1.2x, respectively. In HE, the corresponding ratios are 0.9x, 2.4x, 1.1x, 1.5x, and 4.2x over HiP. However, as illustrated in TABLE V, LC is able to reduce the average bit rate about 51% over BP whereas the average BD-rate between HE and Hip is over 45%. The gap has widened from initial HM versions, where the respective percentages were only 20% and 36% [8].

HM and JM realize all features of the respective standard without optimizations, so they are targeted for research and conformance testing rather than practical real-time decoders. Since HM is currently the only available HEVC decoder, its attainable complexity reduction is here predicted through complexity ratio of JM and an optimized H.264/AVC decoder incorporated in FFMpeg [14]. Conducting the same tests with the optimized H.264/AVC decoder averagely consumes 75% less computational power than JM on a single thread. If the

TABLE IV. THE BEST-CASE TEST SEQUENCES

Seq.	Prof.	QP	ED (%)	IQ/IT (%)	IP (%)	MC (%)	LF (%)	Misc (%)	Total (Mcpf)
SteamLocomotive (1600p)	HE	22	5.7	12.1	1.2	24.4	40.7	15.9	1230.0
		27	2.4	8.9	0.9	31.5	40.3	16.1	932.6
		32	1.4	7.3	0.7	36.0	38.5	16.1	804.0
		37	1.0	6.2	0.6	41.1	34.4	16.7	698.1
	LC	22	4.6	15.4	2.2	40.1	16.3	21.5	750.8
		27	2.8	12.1	1.5	48.2	16.6	18.9	608.6
		32	2.0	9.8	1.1	52.3	16.3	18.6	549.6
		37	1.5	7.8	0.9	55.4	16.1	18.3	517.9
Cactus (1080p)	HE	22	9.8	5.3	1.2	25.8	39.1	18.9	645.4
		27	4.7	4.8	1.1	32.2	37.9	19.4	443.5
		32	3.0	4.5	0.9	36.2	36.5	19.0	368.5
		37	1.9	4.3	0.8	39.3	35.2	18.4	325.1
	LC	22	6.9	6.5	2.2	42.3	16.2	26.0	408.8
		27	4.5	6.9	1.7	47.9	15.7	23.3	298.4
		32	3.3	6.4	1.4	51.3	15.4	22.2	262.0
		37	2.5	5.9	1.1	53.4	15.5	21.6	240.9
BQMall (480p)	HE	22	8.8	3.9	1.2	34.7	30.5	20.8	111.0
		27	5.6	3.2	1.2	38.0	32.2	19.9	89.9
		32	3.7	2.7	1.0	42.1	30.8	19.6	74.7
		37	2.7	2.7	0.9	48.6	25.2	19.9	61.2
	LC	22	6.5	3.9	2.0	49.1	12.7	25.8	77.8
		27	4.8	3.7	1.8	52.7	13.0	24.0	64.2
		32	3.5	3.8	1.5	55.1	13.2	22.9	56.1
		37	2.6	3.3	1.2	57.6	13.0	22.3	51.1
BlowingBubbles (240p)	HE	22	13.5	3.4	1.0	33.6	26.2	22.5	35.7
		27	8.9	2.6	1.1	38.8	26.7	21.8	26.3
		32	6.1	2.4	1.1	45.1	23.1	22.1	19.7
		37	4.0	2.5	1.1	52.2	17.1	23.1	15.3
	LC	22	9.0	3.8	1.5	46.3	10.0	29.5	25.3
		27	6.5	3.5	1.6	51.2	10.8	26.4	19.7
		32	4.6	3.4	1.5	54.4	11.1	25.1	16.1
		37	3.1	3.1	1.2	58.0	10.6	24.0	13.9

equivalent speed-up ratio is assumed between HM and an optimized HEVC decoder, the complexity of HEVC decoding would be below 200 Mcpf at 1080p format (TABLE III). I.e., real-time (30 fps) performance requirement would be around 6 000 M cycles per second. In theory, that complexity would be tackled with 3 GHz dual-core processor and a dual-threaded HEVC decoder.

V. CONCLUSIONS

This paper analyzed the complexity of HEVC reference decoder (HM 3.1) and compared the results with H.264/AVC reference decoder (JM 18.0). In HM, changing settings from LC to HE increases decoding complexity by 42%. The most complex functions of HM are MC and LF, whose respective shares are 50% and 14% in LC as well as 37% and 32% in HE. Under the same QP value, the average complexities of LC and HE are 61% and 87% higher than those of JM BP and JM HiP, respectively. However, HM outperforms JM noticeably in terms of the coding efficiency. The average BD-rate gains of LC over BP and HE over Hip are 51% and 45%, respectively. Assuming that the complexity of HM can be reduced by 75% as in the case of JM, real-time HEVC decoding up to 1080p format could be possible with 3 GHz dual-core processor and a dual-threaded HEVC decoder. The processing technology improvements will further alleviate usage of HEVC standard in the next-generation video products and services.

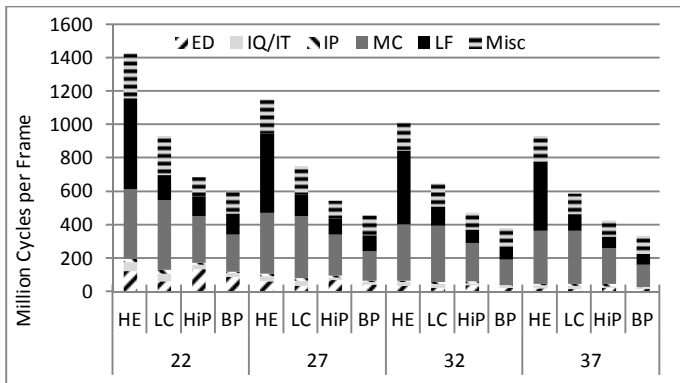


Figure 2a. 1600p sequences

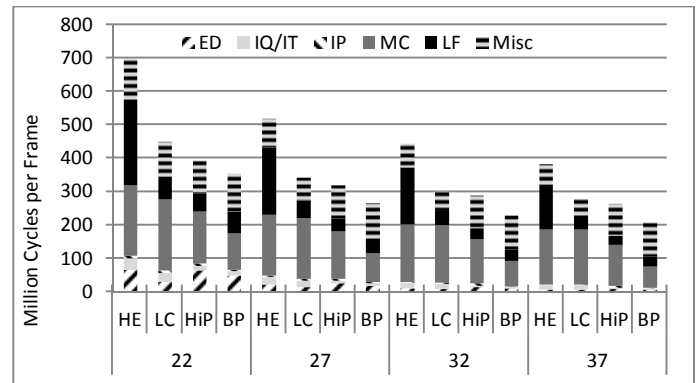


Figure 2b. 1080p sequences

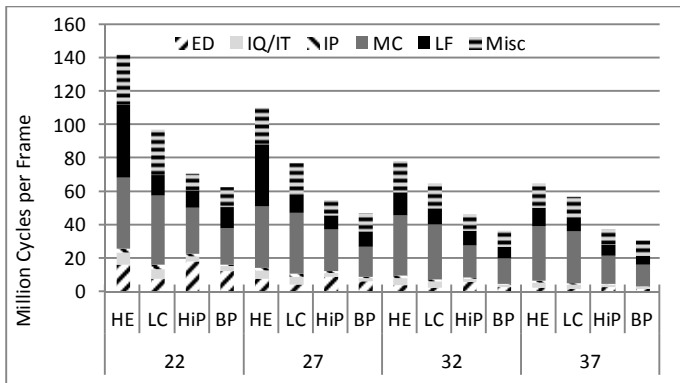


Figure 2c. 480p sequences

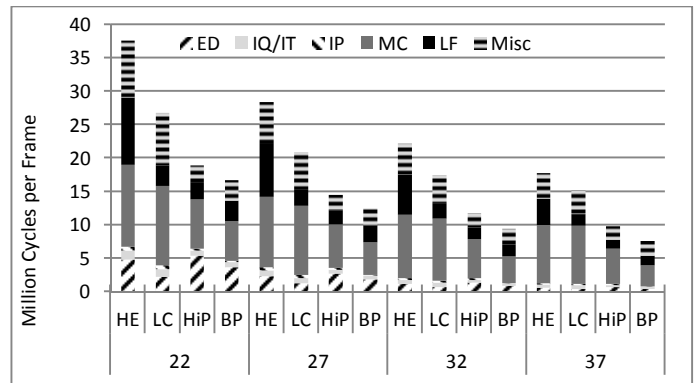


Figure 2d. 240p sequences

Figure 2. Complexity comparison between HEVC (HM 3.1) and H.264/AVC (JM 18.0).

TABLE V. BD-RATE COMPARISON BETWEEN HM AND JM

Format	QP _{HM}	HE vs. HiP		LC vs. BP	
		PSNR (dB)	BD-rate saving (%)	PSNR (dB)	BD-rate saving (%)
1600p	22	41.2	39.5	40.8	42.1
	27	38.4	41.8	38.1	44.8
	32	35.6	44.1	35.3	47.2
	37	32.9	47.3	32.6	51.5
1080p	22	39.5	42.7	39.3	48.6
	27	37.5	50.0	37.3	56.5
	32	35.5	54.4	35.3	60.0
	37	33.3	58.7	33.1	64.2
480p	22	39.7	37.7	39.4	45.5
	27	36.5	41.3	36.3	48.6
	32	33.6	46.0	33.4	51.0
	37	30.8	50.7	30.7	55.9
240p	22	39.3	35.2	39.0	44.4
	27	35.7	39.0	35.5	48.3
	32	32.4	42.1	32.3	50.4
	37	29.6	45.6	29.5	53.1
Average			44.8		50.8

REFERENCES

[1] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," *International Telecommunication Union*, Mar. 2009.

[2] J.-R. Ohm and G. Sullivan, "Vision, applications and requirements for high efficiency video coding (HEVC)," *document N11872*, Daegu, South Korea, Jan. 2011.

[3] F. Bossen *et al.*, "Video coding using a simplified block structure and advanced coding techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, Dec. 2010, pp. 1667-1675.

[4] W. J. Han *et al.*, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, Dec. 2010, pp. 1709-1720.

[5] M. Karczewicz *et al.*, "A hybrid video coder based on extended macroblock sizes, improved interpolation, and flexible motion representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, Dec. 2010, pp. 1698-1708.

[6] D. Marpe *et al.*, "Video compression using nested quadtree structures, leaf merging, and improved techniques for motion representation and entropy coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, Dec. 2010, pp. 1676-1687.

[7] K. Ugur *et al.*, "High performance, low complexity video coding and the emerging HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, Dec. 2010, pp. 1688-1697.

[8] S. Park, J. Park, and B. Jeon, "Report on the evaluation of HM versus JM," *document JCTVC-D181*, Daegu, South Korea, Jan. 2011.

[9] F. Bossen, "Common test conditions and software reference configurations," *document JCTVC-E700*, Geneva, Switzerland, Mar. 2011.

[10] I. K. Kim *et al.*, "Experiments on tools in Working Draft (WD) and HEVC Test Model (HM-3.0)," *document JCTVC-F465*, Torino, Italy, Jul. 2011.

[11] F. Bossen, D. Flynn, and K. Sühring, "AHG Report: Software development and HM software technical evaluation," *document JCTVC-F003*, Torino, Italy, Jul. 2011.

[12] Joint Video Team Reference Software, ver. JM 18.0, Available online: <http://iphome.hhi.de/suehring/tml/>.

[13] T. Wiegand, W. J. Han, B. Bross, J. R. Ohm, and G. J. Sullivan, "WD3: Working Draft 3 of High-Efficiency Video Coding," *document JCTVC-E603*, Geneva, Switzerland, Mar. 2011.

[14] FFmpeg, Available online: <http://www.ffmpeg.org/>.