Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs

Jarno Vanne, *Member, IEEE,* Marko Viitanen, *Member, IEEE,* Timo D. Hämäläinen, *Member, IEEE,* and Antti Hallapuro

***Abstract*—This paper analyzes the rate-distortion-complexity of High Efficiency Video Coding (HEVC) reference video codec (HM) and compares the results with AVC reference codec (JM). The examined software codecs are HM 6.0 using Main Profile (MP) and JM 18.0 using High Profile (HiP). These codes are benchmarked under the all-intra (AI), random access (RA), low-delay *B* (LB), and low-delay *P* (LP) coding configurations. In order to obtain a fair comparison, JM HiP anchor codec has been configured to conform to HM MP settings and coding configurations. The rate-distortion comparisons rely on objective quality assessments, i.e., bit rate differences for equal PSNR. The complexities of HM and JM have been profiled at the cycle level with Intel VTune on Intel Core 2 Duo processor. The coding efficiency of HEVC is drastically better than that of AVC. According to our experiments, the average bit rate decrements of HM MP over JM HiP are 23%, 35%, 40%, and 35% under the AI, RA, LB, and LP configurations, respectively. However, HM achieves its coding gain with a realistic overhead in complexity. Our profiling results show that the average software complexity ratios of HM MP and JM HiP encoders are 3.2× in the AI case, 1.2× in the RA case, 1.5× in the LB case, and 1.3× in the LP case. The respective ratios with HM MP and JM HiP decoders are 2.0×, 1.6×, 1.5×, and 1.4×. This paper also reveals the bottlenecks of HM codec and provides implementation guidelines for future real-time HEVC codecs.**

***Index Terms*—Decoder, encoder, High Efficiency Video Coding (HEVC), HEVC Test Model (HM), rate-distortion-complexity.**

I. Introduction

THE transmission of next-generation video requires coding efficiency that is beyond the capabilities of the current state-of-the-art *AVC (advanced video coding)* standard (ITU-T H.264/ISO MPEG-4 part 10/AVC) [1]. Therefore, MPEG and VCEG have established a *Joint Collaborative Team on Video Coding* (*JCT-VC*) to develop a successor to AVC. This forthcoming international standard is called *HEVC (High Efficiency Video Coding)* [2], [3]. Since 2010, the technical content of the draft standard has been refined from the best-performing initial HEVC proposals [4]–[8]. The *Committee Draft* (*CD*) of HEVC [2] was approved in February 2012 and its *Draft International Standard* (*DIS*) was issued in July 2012. HEVC DIS includes a single profile called *Main Profile* (*MP*) with two tiers (Main and High) and 13 levels [3]. The final standard is planned to be published in early 2013.

HEVC reference codec is called *HEVC Test Model* (*HM*) [9]. In earlier HM versions, the coding tools of HM have been separately specified for *Low Complexity* (*LC*) and *High Efficiency* (*HE*) operation in order to examine the different tradeoffs between coding efficiency and coding complexity [10]. HM 5.0 introduced a separate HE10 for 10-bit operation mode besides HE and LC modes. HM 6.0 [9] represents HEVC CD. Since HM 6.0, the tools of HM have been divided between MP and HE10. Currently, HM 8.0 is the latest version of HM and it represents HEVC DIS. HM testing is recommended to be accomplished according to common test conditions [11] which include four predefined coding configurations: *all-intra* (*AI*), *random access* (*RA*), *low-delay P* (*LP*), and *low-delay B* (*LB*).

The compression performance of HEVC is significantly improved from that of AVC. The evaluations in [12] show that the initial HM versions roughly halve the bit rate over AVC reference encoder (*JM*) [13] with the same subjective visual quality. Under the LP configuration, the HM HE version is reported to achieve 50% bit rate reduction over JM *High Profile* (*HiP*) even with better subjective quality [14].

Although these subjective quality assessments such as the *mean opinion score* (*MOS*) tend to be considered as the most reliable ones, they are cumbersome to organize. Therefore, automatic and repeatable objective quality measures such as *Peak Signal-to-Noise Ratio* (*PSNR*), *Structural SIMilarity* (*SSIM*), and *Perceptual Quality Index* (*PQI*) [15] are typically used when subjective results are not available. PSNR is a simple and the most popular objective measure. It has been shown to yield coherent average results with more sophisticated SSIM and PQI metrics when *rate-distortion* (*RD*) performances of HM and JM are compared [16].

The existing objective quality assessments have focused on PSNR-based RD evaluations [16]–[19] in which HM and JM codecs are compared in terms of *Bjøntegaard delta bit rate* (*BD-rate*) for equal PSNR [20]. However, all these publicly available BD-rate evaluations cover only a subset of the AI, RA, LP, and LB configurations. In addition, most of them consider HM versions prior to 6.0, so their comparisons are limited to previous operating modes of HM such as HE due to

Manuscript received April 16, 2012; revised July 19, 2012; accepted August 22, 2012. Date of publication October 5, 2012; date of current version January 8, 2013. This work was supported in part by the Academy of Finland. This paper was recommended by Associate Editor B. Pesquet-Popescu.

J. Vanne, M. Viitanen, and T. D. Hämäläinen are with the Department of Computer Systems, Tampere University of Technology, Tampere 33101, Finland (e-mail: jarno.vanne@tut.fi; marko.viitanen@tut.fi; timo.d.hamalainen@tut.fi).

A. Hallapuro is with the Nokia Research Center, Tampere 33721, Finland (e-mail: antti.hallapuro@nokia.com).

Digital Object Identifier 10.1109/TCSVT.2012.2223013

the absence of MP. Recently, HM 6.0 has been benchmarked in [18] and HM 7.0 in [19]. According to [18], MP of HM 6.0 can achieve 22%, 33%, and 34% BD-rate savings over JM HiP under the AI, RA, and LB cases, respectively. The corresponding gains of HM 7.0 are reported to be close to those of HM 6.0: 22%, 33%, and 35% [19]. However, these experiments report only BD-rates that cannot illustrate the variations of the delta bit rates of the codecs in the different RD points. BD-rates also deviate a bit from the actual delta bit rates since BD-rates are based on few experimentally specified RD points through which the rest of the considered RD points have been interpolated.

For the time being, the complexity evaluations of the complete HEVC codecs are restricted to runtime comparisons in which consecutive HM versions [21] or HM and JM [22] are benchmarked. The results in [22] are also quite obsolete, since a predecessor of HM 1.0 is benchmarked against JM. The other public complexity assessments focus on HEVC decoders. The profiling results of the HM 4.0 decoder in Intel and ARM processors are shown in [23]. However, the profiling has been conducted on a small test set and the results have been derived from function calls without considering internal complexities of the functions. The profilings in [24]–[26] have been done on platform-specific HM 4.0-based decoders that do not support all HM functions. In addition, the experiments on these proprietary decoders are not reproducible.

Our previous work [27] improves profiling precision by evaluating the HM 3.1 decoder (HE and LE) at the cycle level under a test set that covers the RA configuration. Now, our motivation is to upgrade these results to represent HM 6.0 decoder and extend the test set with the AI, LB, and LP configurations. The complete absence of accurate HEVC encoder assessments gives us reason to do the same profiling with the HM 6.0 encoder too. Fair complexity comparison between HM and JM also requires parameters from detailed RD comparisons not existing in the literature.

In summary, this paper provides a comprehensive *rate-distortion-complexity* (*RDC*) comparison between HM MP and JM HiP codecs under the AI, RA, LP, and LB configurations. The RD comparison is based on the bit rate differences for identical PSNR, whereas cycle-level profiling results have been yielded with Intel VTune™ Amplifier XE 2011 on Intel Core2 Duo E8400 processor. A balanced codec comparison has been accomplished by configuring JM HiP according to HM MP settings. HM has been selected as HEVC codec, because it incorporates all essential HEVC tools and is the only publicly available HEVC codec at the moment. HEVC MP is included in the released HEVC draft standard, so the provided results will serve as a valid platform-independent point of reference for future HEVC codec implementations.

The rest of this paper is organized as follows. Section II presents the main encoding and decoding stages of HEVC codec. Section III describes the setup for the comparative RDC analysis of HM MP and JM HiP. Section IV specifies the bit rate differences between HM and JM. Section V examines the complexities of HM and JM codecs at the cycle level and discusses about practical implementation alternatives for HEVC codecs. Section VI concludes this paper.

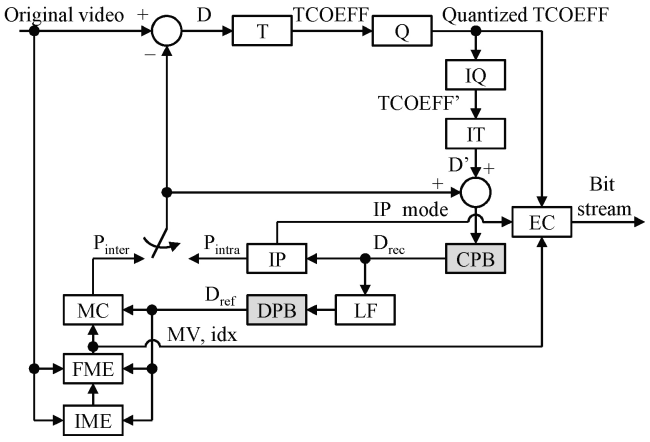

Fig. 1. HEVC encoder model.

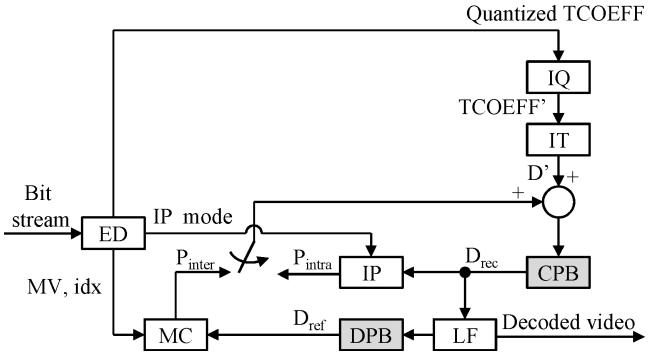

Fig. 2. HEVC decoder model.

II. Overview of HEVC MP Codec

Figs. 1 and 2 depict block diagrams of HEVC encoder and decoder, respectively. From prior video coding standards, HEVC codec adopts a well-known *hybrid video coding scheme* that combines *inter/intra prediction*, *transform coding*, and *entropy coding*. However, the coding structure of HEVC is extended from a traditional *macroblock* (*MB*) concept to an analogous *quadtree scheme* in which the largest *coding unit* (*CU*) can be 16×16, 32×32, or 64×64 luminance pixels. In addition, each CU can be recursively divided into four equally sized CUs until the block granularity is 8×8 pixels. That is, the size of the CU can be defined as $2N \times 2N$ where $N \in \{4, 8, 16, 32\}$ if the maximum hierarchical CU depth of four is applied.

Here, the main focus is on the HEVC MP codec. HEVC MP shares many properties with AVC HiP [1], so the tools unavailable in AVC HiP codec are particularly addressed.

A. Inter Prediction

In inter prediction, CUs at the last level of the CU tree are further divided into one or more rectangular-shaped *prediction units* (*PUs*). For CUs of size $2N \times 2N$, HEVC supports symmetric PUs of size $2N \times 2N$, $2N \times N$, $N \times 2N$, and $N \times N$ (PUs of size 4×4 are disabled). If $N > 4$, HEVC can also

utilize *asymmetric motion partition* (*AMP*) [5] which allows CUs to be split into two asymmetric PUs whose sizes are $2N \times N/2$ and $2N \times 3N/2$ or, alternatively, $N/2 \times 2N$ and $3N/2 \times 2N$.

Luminance motion parameters associated with each PU include *motion vectors* (*MVs*) and corresponding *reference picture/prediction direction indices* (*idxs*). In HEVC, these parameters can be either implicitly derived via motion merging (*merge mode*) or they can be explicitly estimated through normal inter prediction (*inter mode*) [7], [10]. In both cases, chrominance MVs are derived from luminance ones.

The merge mode infers motion parameters for the processed PU from spatially and temporally adjacent inter coded PUs. HEVC MP specifies four spatial merge candidates (neighboring PUs) and one temporal merge candidate (temporally colocated PU). If less than five distinct spatiotemporal candidates are available, more candidates are artificially generated from the existing ones so that the number of final merge candidates reaches five. The costs of these five candidates are computed and the best one of them is chosen. Merge mode is skipped if none of the candidates is available.

In inter mode, the motion parameters are obtained through *motion estimation* (*ME*) that includes *integer ME* (*IME*) and *fractional ME* (*FME*) stages (see Fig. 1). ME accesses data from a *decoded picture buffer* (*DPB*) which contains the previously reconstructed *reference pictures* (D_{ref}). The first phase of ME is IME that searches for the best candidates for the processed PU from D_{ref}. HEVC enhances IME through *advanced MV prediction* (*AMVP*) [5], [10] that derives the best *MV predictor* (*MVP*) from two spatially and one temporally adjacent MVP candidates. The selection process of the best MVP follows that of motion merge, except that the number of final spatiotemporal MVP candidates is two. IME delivers integer-pixel accurate MVs and Idxs of the best matches to FME that refines luminance MVs to 1/4-pixel accuracy and chrominance MVs to 1/8-pixel accuracy. HEVC uses 8-tap separable *interpolation* (*IPOL*) filter for 1/4-pixel luminance samples and 4-tap separable IPOL filter for 1/8-pixel chrominance samples. Both filters have been upgraded from those in AVC.

Motion compensation (*MC*) produces *inter predictions* (P_{inter}) for PUs by addressing DPB with MVs and Idxs. If the encoder operates in inter mode, a *prediction residual* (D) is computed by subtracting P_{inter} from the processed original CU. However, if CU is encoded as skip mode, no D is computed, only PUs of size $2N \times 2N$ are allowed, and motion parameters are derived through merge mode.

B. Intra Prediction

In intra prediction, PUs may take the size of $2N \times 2N$. In addition, intra coded PUs of size $N \times N$ are supported when $N = 4$. The unified intra prediction coding tool of HEVC increases *IP modes* over AVC by supporting 35 IP modes (DC, planar, and 33 angular IP modes) for each PU size.

An *intra prediction* (*IP*) stage computes intra prediction (P_{intra}) for the processed PU by accessing a *current picture buffer* (*CPB*) that contains previously reconstructed blocks of the *current picture* (D_{Rec}). In intra mode, the encoder computes D by subtracting P_{intra} from the original CU.

C. Transform and Quantization

For transform and quantization, HEVC specifies *transform unit* (*TU*), whose shape depends on PU. HEVC MP supports only square-shaped TUs of size 4×4, 8×8, 16×16, and 32×32 pixels. Multiple TUs inside a single CU can be arranged in a quadtree structure whose maximum depth is three. TUs can also cross boundaries of inter coded PUs but not boundaries of intra coded PUs.

A *transform* (*T*) stage converts spatial domain D into *transform domain coefficients* (*TCOEFFs*) after which TCOEFFs are quantized in a *quantization* (*Q*) stage. HEVC utilizes integer *Discrete Sine Transform* (*DST*) for intra coded 4×4 luminance TUs and integer *Discrete Cosine Transform* (*DCT*) for the other TUs [3]. All transform matrices have been upgraded from AVC with added precision in the integer scale.

The decoding path of the encoder uses *inverse quantization* (*IQ*) and *inverse transform* (*IT*) stages to dequantize and convert *Quantized TCOEFFs* back to spatial domain D (D'). D_{Rec} is then yielded by adding $P_{\text{inter}}/P_{\text{intra}}$ to D'.

D. Entropy Coding

In parallel with the decoding path, an *entropy coding* (*EC*) stage converts MVs, Idxs, quantized TCOEFFs, and other syntax elements to binary codewords which are multiplexed together to a bit stream. In HEVC, the used EC technique is *context-adaptive binary arithmetic coding* (*CABAC*).

E. Loop Filtering

A *loop filtering* (*LF*) stage filters the distortions and visible CU/PU/TU borders from the picture. The LF stage of HEVC MP contains two sequential in-loop filters: *deblocking filter* (*DF*) and *sample-adaptive offset* (*SAO*).

F. Decoding

In the decoder side (see Fig. 2), an *entropy decoder* (*ED*) stage extracts CABAC-coded binary codewords from the input bit stream and converts them back to original syntax elements including IP mode, quantized TCOEFFs, MVs, and Idxs. The IQ and IT stages are duplicated from the encoder. They dequantize and transform quantized TCOEFFs back to D'. IP produces P_{intra} according to IP mode and MC yields P_{inter} as in the encoder. The decoder composes D_{Rec} by adding D' together with P_{intra} in intra mode or with P_{inter} in inter mode. It produces decoded video by filtering D_{Rec} with DF and SAO.

III. Analysis Setup

Table I tabulates the main coding options of HM MP and JM HiP codecs. During the experiments performed for this paper, HM 6.0 [9] was the latest available version of HM. Contrary to MP of HM 8.0 (and HM 7.0), HM 6.0 excludes AMP from the inter coding tools of MP. However, the effect of AMP on RD performance is not significant according to the overal RD results with [19] and without [18] AMP. From the

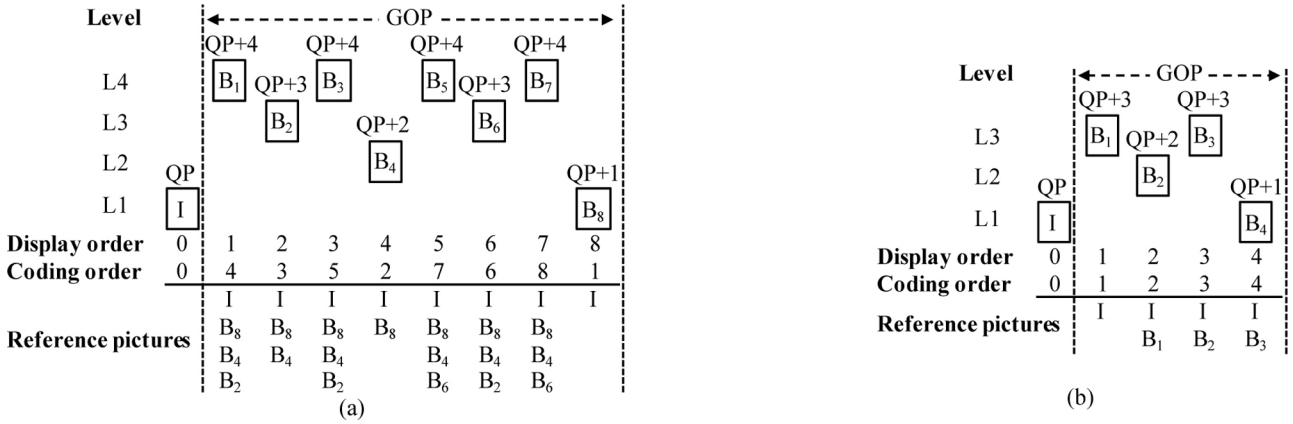

Fig. 3. Hierarchical coding structures of the RA and LB configurations. (a) RA configuration. (b) LB configuration.

TABLE I
CODING OPTIONS OF HM MP AND JM HIP CODECS

Coding option	HM MP	JM HiP
Internal bit depth	8	8
Sizes of CUs	64 × 64, 32 × 32, 16 × 16, 8 × 8	16 × 16, 8 × 8
Sizes of TUs	32 × 32, 16 × 16, 8 × 8, 4 × 4	8 × 8, 4 × 4
Sizes of PUs	64 × 64, 64 × 32, 32 × 64, 32 × 32, 32 × 16, 16 × 32, 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8	16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, 4 × 4
Entropy coding	CABAC	CABAC
Loop filtering	DF, SAO	DF
IME algorithm	EPZS	EPZS
Search range	[−64, +64]	[−64, +64]
No. of reference pictures	4	4
IME metric	SAD	SAD
FME metric	SATD	SATD
Mode decision metric	SATD	SATD
RDO	Enabled	Enabled
RDOQ	Enabled	Enabled
No. of QPs in RDOQ	1	1

RD analysis point of view, the other inconsistencies between MPs of HM 6.0 and HM 8.0 are also expected to be marginal.

Our experiments rely on the default configuration file of HM 6.0 according to which the configuration file of JM 18.0 [13] has been parametrized (JM software has not been modified). In both codecs, the nonnormative IME is realized with *enhanced predictive zonal search* (*EPZS*) [28] that uses four reference pictures and the search range of [−64, +64] both horizontally and vertically. IME relies on *Sum of Absolute Differences* (*SAD*) as a similarity criterion for distortion computation, whereas FME and *coding mode decision* (*MD*) are parametrized to use *Sum of Absolute Transformed Differences* (*SATD*) criterion. Contrary to our previous work [27], both codecs also support *RD optimized* (*RDO*) *mode decision* and *RDO quantization* (*RDOQ*) with a single tested *quantization parameter* (*QP*).

A. Test Conditions

HM uses QP values of 22, 27, 32, and 37 according to common test conditions [11]. QPs of JM have been experimentally accommodated to QPs of HM by streamlining PSNRs of the codecs. In our experiments, HM and JM have been analyzed under the AI, RA, LB, and LP configurations using the coding structures adopted mainly from [10]:

For the AI condition, pictures are coded as intra (*I*) pictures in display order without temporal references and QP offsets.

For the RA condition, *I* picture is inserted roughly at one second intervals and the other pictures are coded as *B* pictures. The RA configuration exploits four-layer (L1, L2, L3, and L4) hierarchical coding structure in which the *GOP* (*Group of Pictures*) size is eight. Fig. 3(a) depicts this coding structure for the first nine pictures (I, B_1, ..., B_8) of the sequence. The coding order of the pictures in the GOP is B_8, B_4, B_2, B_1, B_3, B_6, B_5, B_7 and they are located at layers L1, L2, L3, L4, L4, L3, L4, L4, respectively. *I* pictures are coded with original QP, whereas a QP offset of each *B* picture is equal to its layer index. Fig. 3(a) also lists the prediction dependencies between the pictures. For example, B_1 uses I, B_8, B_4, and B_2 as references.

The LB condition uses three-level hierarchical coding structure with the GOP size of four. Fig. 3(b) depicts this coding structure for the first five pictures of the sequence. The pictures in a GOP are coded in a display order as B_1, B_2, B_3, B_4 at layers L3, L2, L3, and L1, respectively. Only the first picture of the sequence is *I* picture and the others are *B* pictures. QP offsets are derived as in the RA condition. The coding structure used in the LP condition resembles that of the LB case expect that *B* pictures are replaced with *P* pictures.

B. Test Setup for Rate-Distortion Comparison

Table II lists the 8-bit test sequences recommended by common test conditions [11] for the AI, RA, LB, and LP configurations. This test set is also used in our RD comparisons between HM MP and JM HiP. Two 10-bit sequences included in [11] have been excluded from our test set, since they are beyond the capabilities of JM HiP.

The RD performances of HM MP and JM HiP have also been compared as a function of the resolution. This comparison has been carried out with Class *A* sequences starting from their original (uncropped) resolutions: *Traffic* (4096 × 2048, the first 150 frames) and *PeopleOnStreet* (3840 × 2160, 150 frames). These two sequences have been scaled down to create the formats that represent the Classes *A*–*E*. The scaling has been performed with a 12-tap nonnormative downsampling filter of *Joint Scalable Video Model* (*JSVM*) software [29].

TABLE II
TEST SEQUENCES

Class	Format	Sequence	No. of frames	Frame rate	Condition AI	RA	LB	LP
A	2560 × 1600 (1600p)	*Traffic*	150	30 fps	X	X		
		PeopleOnStreet	150	30 fps	X	X		
B	1920 × 1080 (1080p)	*Kimono*	240	24 fps	X	X	X	X
		ParkScene	240	24 fps	X	X	X	X
		Cactus	500	50 fps	X	X	X	X
		BQTerrace	600	60 fps	X	X	X	X
		BasketballDrive	500	50 fps	X	X	X	X
C	832 × 480 (WVGA)	*RaceHorses*	300	30 fps	X	X	X	X
		BQMall	600	60 fps	X	X	X	X
		PartyScene	500	50 fps	X	X	X	X
		BasketballDrill	500	50 fps	X	X	X	X
D	416 × 240 (WQVGA)	*RaceHorses*	300	30 fps	X	X	X	X
		BQSquare	600	60 fps	X	X	X	X
		BlowingBubbles	500	50 fps	X	X	X	X
		BasketballPass	500	50 fps	X	X	X	X
E	1280 × 720 (720p)	*FourPeople*	600	60 fps	X		X	X
		Johnny	600	60 fps	X		X	X
		KristenAndSara	600	60 fps	X		X	X
F	WVGA	*BasketballDrillText*	500	50 fps	X	X	X	X
	1024 × 768	*ChinaSpeed*	500	30 fps	X	X	X	X
	720p	*SlideEditing*	300	30 fps	X	X	X	X
		SlideShow	500	20 fps	X	X	X	X

Since the aspect ratios of the original formats have been kept constant, the widths of the downsampled resolutions differ a bit from the ones in Table II.

In this paper, the bit rate differences between HM MP and JM HiP have been examined as a function of $PSNR_{AVG}$ that is a weighted average of luminance (PSNR_Y) and chrominance (PSNR_U and PSNR_V) PSNR components [17], [30]. All involved test sequences (see Table II) are in 4:2:0 color format, for which $\mathrm{PSNR}_{\mathrm{AVG}}$ is computed as

$$\mathrm{PSNR}_{\mathrm{AVG}} = \frac{6 \times \mathrm{PSNR}_Y + \mathrm{PSNR}_U + \mathrm{PSNR}_V}{8}. \quad (1)$$

Since $\mathrm{PSNR}_{\mathrm{AVG}}$ also takes the impact of the chrominance components into account, it is supposed to provide more reliable results than the conventional PSNR_Y metric in the cases when the luminance and chrominance components have dissimilar RD behaviors [30].

The RD points of JM anchors ($\mathrm{RD}_{\mathrm{JM}}$) have been obtained for RD comparisons by encoding the involved test sequences with 24 different QP ($\mathrm{QP}_{\mathrm{JM}}$) values ranging from 17 to 40 (delta $\mathrm{QP}_{\mathrm{JM}}$ = 1). The corresponding sequence-specific RD points of HM anchors ($\mathrm{RD}_{\mathrm{HM}}$) represent $\mathrm{QP}_{\mathrm{HM}}$ values of 22, 27, 32, and 37 and their $\mathrm{PSNR}_{\mathrm{AVG}}$ values have been accommodated to the associated $\mathrm{RD}_{\mathrm{JM}}$ curve. Fig. 4(a) depicts the principle of locating the $\mathrm{RD}_{\mathrm{JM}}$ points of interest from the $\mathrm{RD}_{\mathrm{JM}}$ curve. For each $\mathrm{RD}_{\mathrm{HM}}$ point ($\cdot$), the comparable $\mathrm{RD}_{\mathrm{JM}}$ point has been interpolated from the four nearest $\mathrm{RD}_{\mathrm{JM}}$ anchor points (+) around the $\mathrm{PSNR}_{\mathrm{AVG}}$ value of interest. In Fig. 4(a), the circled groups of $\mathrm{RD}_{\mathrm{JM}}$ anchor points are the ones used in the interpolations. The $\mathrm{RD}_{\mathrm{JM}}$ points of interest are pointed with the arrows (the end points of the $\mathrm{RD}_{\mathrm{JM}}$ curve not drawn). The interpolations have been performed with a third-order polynomial function adopted from [20].

TABLE III
PROFILING PLATFORM FOR COMPLEXITY ANALYSIS

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

Using four local interpolations improves the interpolation accuracy over the case where a single interpolation curve is fitted over the whole range. Fig. 4(b) visualizes the latter case where the $\mathrm{RD}_{\mathrm{JM}}$ anchor points represent $\mathrm{QP}_{\mathrm{JM}}$ values of 22, 27, 32, and 37 (delta $\mathrm{QP}_{\mathrm{JM}}$ = 5). With the applied test set (see Table II), decreasing the granularity from delta $\mathrm{QP}_{\mathrm{JM}}$ = 5 to delta $\mathrm{QP}_{\mathrm{JM}}$ = 1 improves the bit rate estimates of individual $\mathrm{RD}_{\mathrm{JM}}$ points around 1% on average. This improvement is due to interpolation mismatch that can be identified by interpolating the missing $\mathrm{RD}_{\mathrm{JM}}$ anchor points in delta $\mathrm{QP}_{\mathrm{JM}}$ = 5 case and comparing the interpolation outcomes with the actual anchor points available in delta $\mathrm{QP}_{\mathrm{JM}}$ = 1 case. Here, the interpolation accuracy has only been examined with $\mathrm{QP}_{\mathrm{JM}}$ values from 21 to 38 to avoid overweighting the importance of rarely used end points whose interpolation errors are higher.

C. Test Setup for Complexity Profiling

Table III tabulates the profiling platform for the codecs. Our profiling environment is composed of two of these identical processor platforms. During the analysis, a codec under test has been the only software running to reduce noise caused by other computer processes on the results. Hence, only a single core per Core 2 Duo processor has been used. SIMD extensions (MMX/SSE) of the processors have not been exploited in order to maintain platform independency.

The analysis relies on Intel VTune profiler which is able to report estimated cycle counts for each function of the codecs. Cycle-level profiling also considers internal complexities of the functions so it is more reliable than the analysis monitoring function calls only. This complexity analysis reuses the test set of RD comparison (see Table II) but excludes Class *F* due to its heterogeneous sequence resolutions.

HM profiling has been conducted with $\mathrm{QP}_{\mathrm{HM}}$ values of 22, 27, 32, and 37. JM profiling uses the sequence-specific $\mathrm{QP}_{\mathrm{JM}}$ values that have been accommodated to associated $\mathrm{QP}_{\mathrm{HM}}$ values during the RD comparison. By that way, the profiling of HM and JM codecs is performed with similar $\mathrm{PSNR}_{\mathrm{AVG}}$ values and the complexity overhead of HM can be better mapped to its bit rate gains.

HM MP and JM HiP decoder configurations have been run ten times with the same test set and the reported values are means of the outcomes of these test passes. The average deviation of a single outcome is around 2% among these test passes. HM and JM encoders have been run only twice to save profiling time. The reliability of the average encoder results is estimated to be at the same level as with the decoder profiling.

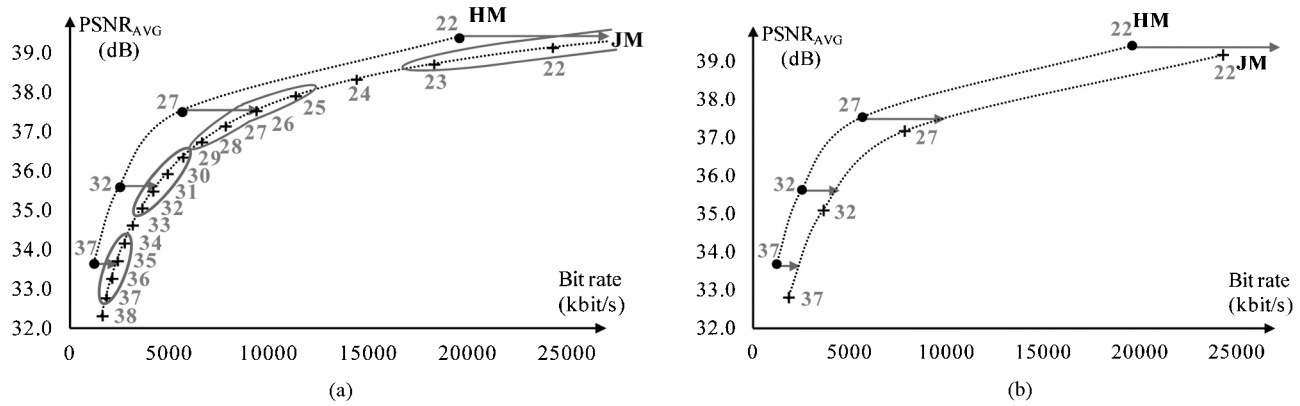

Fig. 4. Locating the RD_{JM} points of interest from the RD_{JM} curve (*Cactus* test sequence under the LB configuration). (a) Delta QP_{JM} = 1. (b) Delta QP_{JM} = 5.

TABLE IV
RELATIONSHIP OF SEQUENCE-SPECIFIC QP SETTINGS BETWEEN HM 6.0 AND JM 18.0

	AI				RA				LB				LP			
Sequence | QP_{HM}	**22**	**27**	**32**	**37**	**22**	**27**	**32**	**37**	**22**	**27**	**32**	**37**	**22**	**27**	**32**	**37**
Traffic	23	28	33	38	21	26	31	35	–	–	–	–	–	–	–	–
PeopleOnStreet	23	28	33	38	22	27	31	36	–	–	–	–	–	–	–	–
Kimono	23	28	33	38	21	26	30	34	22	26	31	35	23	28	33	38
ParkScene	23	28	33	38	21	26	31	35	22	27	31	36	23	28	33	37
Cactus	23	28	33	38	21	26	31	35	22	27	31	36	23	28	33	37
BQTerrace	23	28	33	38	22	25	30	35	22	26	31	35	24	28	32	37
BasketballDrive	23	28	33	38	21	26	30	34	22	26	31	35	23	28	33	38
RaceHorses	23	28	33	38	22	26	31	35	22	27	32	36	23	29	34	38
BQMall	23	28	33	38	21	26	31	35	22	27	31	36	23	28	33	37
PartyScene	23	28	33	38	21	26	31	36	22	27	31	36	24	28	33	38
BasketballDrill	23	28	33	38	21	26	31	35	22	26	31	35	22	27	32	37
RaceHorses	23	28	33	38	22	26	31	35	22	27	32	36	23	29	34	38
BQSquare	23	28	33	38	21	26	30	35	22	26	31	36	23	28	33	37
BlowingBubbles	23	28	33	38	21	26	31	36	22	27	32	37	23	28	33	38
BasketballPass	23	28	33	38	22	26	31	35	22	27	32	36	23	29	34	38
FourPeople	23	28	33	38	–	–	–	–	22	26	31	36	23	28	32	37
Johnny	23	28	32	37	–	–	–	–	21	25	30	34	23	27	31	36
KristenAndSara	23	28	33	38	–	–	–	–	22	26	31	35	23	27	32	36
BasketballDrillText	23	28	33	38	21	26	31	35	22	26	31	35	23	28	32	37
ChinaSpeed	21	27	33	38	22	26	31	35	22	27	31	35	23	28	33	38
SlideEditing	19	26	32	37	20	25	30	35	21	26	30	35	21	26	31	36
SlideShow	19	26	32	37	20	25	30	34	21	26	31	35	22	27	32	36
Average QP_{JM}	23	28	33	38	21	26	31	35	22	26	31	36	23	28	33	37

IV. RD COMPARISON OF HM MP AND JM HIP CODECS

Table IV tabulates a sequence-specific relationship between QP_{JM} and QP_{HM} settings of HM MP and JM HiP codecs when QP_{HM} values are set to 22, 27, 32, and 37. Among the four QP_{JM} values involved in the comparable RD_{JM} point interpolation, the closest one that yields lower $PSNR_{AVG}$ value than the respective RD_{HM} point is reported. As a result, all listed QP_{JM} values represent lower $PSNR_{AVG}$ values than the comparable QP_{HM} values do.

Table V reports the bit rate savings of HM MP over JM HiP for identical $PSNR_{AVG}$ values. For each sequence, the bit rate savings per four individual QP_{HM} values (Δ bit rate/QP_{HM}) and the BD-rates are tabulated. The Δ bit rate/QP_{HM} values have been yielded as in Fig. 4(a) and the BD-rates have been computed using the RD points shown in Fig. 4(b).

The averages of four sequence-specific Δ bit rate/QP_{HM} values deviate around 1 percentage points (pps) from the respective BD-rates. In addition, Δ bit rate/QP_{HM} values are able to illustrate the variation of the Δ bit rate along the RD curves. At QP_{HM} = 22, the average deviation of the sequence-specific Δ bit rate/QP_{HM} and BD-rate values is almost 7 pps (from -35 pps to 18 pps). The respective variations are 2 pps (from -6 pps to 2 pps) at QP_{HM} = 27, 2 pps (from -3 pps to 10 pps) at QP_{HM} = 32, and 6 pps (-4 pps to 19 pps) at QP_{HM} = 37.

The overall bit rate savings of HM MP over JM HiP are summarized in the last rows of Table V. Under the AI case, the average bit rate reduction of HM (Average/condition) is 23% with a sequence-specific variation of 11–38%. The respective bit rate savings under the RA, LB, and LP cases are 35% (21%–53%), 40% (21%–69%), and 35% (16%–63%).

Compared to [18], the average BD-rates reported here are 1, 2, and 6 pps higher in the AI, RA, and LB cases, respectively. The difference is caused by the stronger AVC anchor (JM 18.3) used in [18].

Table VI tabulates the corresponding overall results when $PSNR_{AVG}$ metric is replaced with a conventional $PSNR_Y$ metric, i.e., the overall Δ bit rate/QP_{HM} values and BD-rates are reported for the equal $PSNR_Y$ values. Although replacing $PSNR_{AVG}$ metric with $PSNR_Y$ metric would cause an average deviation around 1 pps for the sequences-specific results, the average results per coding condition (Average/condition) in Table VI are converged close to those in Table V.

TABLE V
SEQUENCE-SPECIFIC AND OVERALL BIT-RATE SAVINGS OF HM 6.0 OVER JM 18.0 FOR EQUAL $PSNR_{AVG}$ VALUES

Sequence	AI					RA					LB					LP				
	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-
	22	27	32	37	rate	22	27	32	37	rate	22	27	32	37	rate	22	27	32	37	rate
Traffic	20%	23%	23%	23%	23%	34%	37%	40%	43%	39%	–	–	–	–	–	–	–	–	–	–
PeopleOnStreet	21%	22%	22%	22%	22%	23%	24%	25%	29%	25%	–	–	–	–	–	–	–	–	–	–
Kimono	27%	29%	29%	29%	29%	42%	42%	46%	53%	46%	36%	37%	44%	54%	42%	29%	31%	35%	43%	34%
ParkScene	15%	17%	19%	20%	18%	34%	33%	36%	43%	36%	33%	34%	40%	51%	38%	29%	32%	37%	45%	34%
Cactus	19%	24%	26%	28%	24%	33%	38%	39%	43%	39%	32%	39%	42%	47%	41%	26%	36%	40%	44%	37%
BQTerrace	12%	21%	25%	29%	21%	23%	48%	50%	52%	48%	22%	57%	63%	69%	56%	18%	38%	54%	63%	44%
BasketballDrive	22%	29%	32%	34%	29%	34%	42%	46%	52%	45%	34%	43%	47%	55%	46%	27%	35%	40%	46%	37%
RaceHorses	16%	18%	20%	24%	19%	21%	27%	32%	40%	30%	21%	28%	33%	40%	30%	16%	23%	25%	31%	23%
BQMall	19%	21%	21%	21%	20%	32%	33%	36%	40%	35%	32%	33%	37%	43%	36%	26%	30%	34%	40%	32%
PartyScene	11%	12%	13%	15%	13%	31%	31%	32%	34%	32%	39%	43%	44%	45%	43%	23%	39%	42%	46%	39%
BasketballDrill	27%	32%	33%	34%	32%	35%	37%	40%	44%	39%	40%	42%	46%	50%	44%	37%	41%	44%	48%	42%
RaceHorses	17%	19%	22%	24%	20%	24%	26%	29%	36%	28%	24%	26%	29%	35%	28%	19%	21%	24%	27%	22%
BQSquare	12%	13%	14%	15%	14%	40%	43%	42%	42%	42%	44%	57%	59%	59%	57%	30%	48%	55%	55%	48%
BlowingBubbles	12%	14%	15%	16%	14%	27%	28%	28%	31%	28%	32%	34%	36%	37%	35%	26%	33%	36%	41%	34%
BasketballPass	20%	23%	24%	24%	23%	25%	28%	32%	36%	30%	25%	28%	32%	37%	30%	21%	25%	28%	30%	25%
FourPeople	23%	24%	23%	22%	24%	–	–	–	–	–	32%	33%	35%	41%	35%	28%	30%	35%	40%	33%
Johnny	30%	35%	38%	38%	36%	–	–	–	–	–	50%	58%	58%	61%	58%	38%	49%	52%	54%	51%
KristenAndSara	27%	29%	31%	32%	30%	–	–	–	–	–	36%	43%	49%	56%	48%	33%	40%	46%	51%	43%
BasketballDrillText	25%	28%	28%	29%	28%	34%	36%	39%	42%	38%	39%	43%	46%	49%	44%	35%	40%	45%	48%	42%
ChinaSpeed	27%	21%	18%	17%	19%	23%	25%	30%	36%	28%	23%	27%	33%	44%	31%	20%	25%	33%	44%	29%
SlideEditing	35%	19%	14%	12%	16%	27%	23%	22%	21%	23%	28%	26%	27%	27%	28%	28%	27%	25%	23%	26%
SlideShow	38%	30%	26%	26%	28%	31%	29%	30%	32%	30%	29%	32%	36%	41%	35%	29%	32%	36%	40%	34%
Minimum	11%	12%	13%	12%	13%	21%	23%	22%	21%	23%	21%	26%	27%	27%	28%	16%	21%	24%	23%	22%
Maximum	38%	35%	38%	38%	36%	42%	48%	50%	53%	48%	50%	58%	63%	69%	58%	38%	49%	55%	63%	51%
Average	22%	23%	23%	24%	23%	30%	33%	35%	39%	35%	32%	38%	42%	47%	40%	27%	34%	38%	43%	35%
Average/condition	23%					35%					40%					35%				

TABLE VI
OVERALL BIT-RATE SAVINGS OF HM 6.0 OVER JM 18.0 FOR EQUAL $PSNR_{Y}$ VALUES

	AI					RA					LB					LP				
	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-
	22	27	32	37	rate	22	27	32	37	rate	22	27	32	37	rate	22	27	32	37	rate
Minimum	11%	13%	14%	13%	13%	20%	25%	22%	20%	23%	19%	26%	27%	26%	27%	15%	20%	22%	23%	21%
Maximum	42%	34%	39%	37%	36%	44%	49%	52%	54%	48%	48%	57%	65%	68%	58%	38%	48%	55%	62%	50%
Average	22%	24%	24%	24%	23%	30%	34%	36%	38%	35%	33%	39%	43%	46%	41%	27%	34%	38%	42%	36%
Average/condition	23%					35%					41%					35%				

As shown in Table V, the bit rate gap between HM and JM increases together with QP value. Incrementing QP_{HM} value from 22 to 37 increases the average Δ bit rate by about 9 pps in the RA case, 15 pps in the LB case, and 16 pps in the LP case. However, in the AI configuration the Δ bit rate remains almost the same with different QP_{HM} values.

Table VII tabulates the bit rate gain of HM MP over JM HiP as a function of the resolution. Among the evaluated two sequences, the average bit rate savings of HM MP are around 11 pps (from 12% to 23%) higher in the AI condition when the resolution is incremented from the lowest to the highest one. The respective increments under the RA, LB, and LP conditions are 14 pps, 17 pps, and 14 pps. In all these cases, the coding efficiency of HM MP continues to grow faster than that of JM HiP also beyond the resolutions involved in [11].

The coding gain of HEVC MP codec is a result of its extended coding structure and upgraded coding tools. Supporting large CU, PU, and TU sizes with content-adaptive block partitioning scheme is a key HEVC technique that can be efficiently adjusted between large homogeneous regions and highly textured areas of the picture. As shown in Table VII, the benefits of the extended coding structure are emphasized with higher resolutions. Tool-level enhancements of HEVC are particularly focused in inter and intra prediction in which the most important tools are advanced intra prediction, more accurate IPOL, motion merging, and AMVP.

V. COMPLEXITY ANALYSIS

Tables VIII and IX tabulate the sequence-specific complexity results of HM encoder and decoder, respectively. The absolute complexities are reported as *million cycles per frame* (*Mcpf*) and the complexity distribution among the main coding stages are tabulated as percentages. In both cases, only the sequences with maximum and minimum cycle counts are reported for each format. These corner cases have been resolved from the sums of the sequence-specific complexities involved in the AI, RA, LB, and LP configurations. Therefore, the reported values may deviate from the maximum/minimum cycle counts in individual test cases.

A. Complexity Analysis of HM MP Encoder

The most complex stages of the encoder are IME, FME/MD, IP, T/Q/IQ/IT, and EC (see Table VIII). Allocating SATD operations between the FME and MD stages would require HM source code modifications, so they are combined in

TABLE VII

BIT-RATE SAVINGS OF HM 6.0 OVER JM 18.0 AS A FUNCTION OF THE RESOLUTION

		AI					RA					LB					LP				
		Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-	Δ bit rate/QP_{HM}				BD-
	Sequence	22	27	32	37	rate	22	27	32	37	rate	22	27	32	37	rate	22	27	32	37	rate
Traffic	4096 × 2048	21%	23%	25%	24%	23%	34%	38%	41%	45%	40%	36%	42%	46%	51%	44%	29%	38%	43%	49%	40%
	3200 × 1600	23%	24%	25%	23%	23%	36%	36%	38%	41%	38%	35%	39%	42%	47%	41%	28%	35%	40%	46%	37%
	2160 × 1080	19%	20%	23%	20%	20%	34%	33%	34%	37%	34%	37%	38%	38%	41%	38%	30%	35%	38%	42%	36%
	1440 × 720	16%	17%	22%	18%	18%	32%	31%	31%	33%	31%	36%	35%	34%	36%	35%	29%	33%	36%	38%	34%
	960 × 480	15%	16%	21%	17%	16%	31%	30%	30%	32%	30%	33%	33%	31%	33%	32%	28%	34%	34%	37%	33%
	480 × 240	10%	11%	18%	14%	12%	17%	20%	23%	26%	21%	19%	25%	29%	29%	26%	13%	23%	34%	33%	25%
PeopleOnStreet	3840 × 2160	21%	23%	25%	24%	23%	22%	25%	26%	30%	26%	22%	23%	25%	29%	24%	17%	19%	21%	25%	20%
	2840 × 1600	38%	24%	25%	23%	25%	23%	24%	25%	28%	24%	22%	20%	23%	26%	22%	17%	18%	21%	22%	19%
	1920 × 1080	19%	20%	23%	21%	20%	22%	23%	24%	26%	24%	19%	19%	21%	23%	20%	16%	17%	19%	21%	18%
	1280 × 720	18%	19%	22%	20%	19%	21%	22%	24%	26%	23%	15%	16%	18%	21%	17%	12%	13%	15%	19%	14%
	848 × 480	15%	17%	21%	18%	17%	20%	21%	23%	25%	22%	11%	12%	15%	18%	14%	9%	11%	12%	16%	11%
	424 × 240	11%	11%	14%	14%	12%	11%	14%	17%	22%	15%	6%	6%	10%	15%	8%	4%	4%	9%	14%	5%

TABLE VIII

COMPLEXITY DISTRIBUTION OF HM MP ENCODING: THE WORST-CASE AND THE BEST-CASE TEST SEQUENCES

			AI							RA							LB							LP						
Format	Complexity	QP	IME	FME/MD	IP	T/Q/IQ/IT	EC	Misc	Total	IME	FME/MD	IP	T/Q/IQ/IT	EC	Misc	Total	IME	FME/MD	IP	T/Q/IQ/IT	EC	Misc	Total	IME	FME/MD	IP	T/Q/IQ/IT	EC	Misc	Total
	(Sequence)		(%)	(%)	(%)	(%)	(%)	(%)	(Mcpf)	(%)	(%)	(%)	(%)	(%)	(%)	(Mcpf)	(%)	(%)	(%)	(%)	(%)	(%)	(Mcpf)	(%)	(%)	(%)	(%)	(%)	(%)	(Mcpf)
1600p	MAX (*PeopleOnStreet*)	22	0	7	22	44	14	13	115 459	21	37	3	21	8	10	427 769	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		27	0	8	24	41	12	15	98 975	22	43	2	17	6	11	367 111	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		32	0	9	27	37	11	16	87 121	23	47	1	14	4	11	330 324	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		37	0	10	29	34	10	17	78 767	23	51	1	11	4	11	306 501	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	MIN (*Traffic*)	22	0	7	22	45	14	13	113 509	14	50	2	17	6	11	310 362	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		27	0	8	25	40	12	15	96 657	15	56	1	12	4	12	272 649	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		32	0	9	28	37	11	16	85 567	16	59	1	9	3	12	255 033	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		37	0	10	29	34	10	17	77 447	16	60	1	8	3	12	245 585	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1080p	MAX (*BasketballDrive*)	22	0	8	21	47	12	13	70 492	20	49	2	17	4	7	281 236	24	48	2	15	4	7	405 559	26	35	2	22	6	8	285 780
		27	0	10	24	42	9	15	58 456	20	56	1	12	3	8	243 046	25	55	1	10	2	7	349 813	27	42	2	16	4	9	232 434
		32	0	11	28	35	8	17	49 659	19	60	1	10	2	8	224 359	24	59	1	8	2	7	319 478	26	48	1	12	3	9	200 639
		37	0	11	28	36	7	17	48 597	18	63	1	8	2	8	210 782	22	62	0	7	1	7	301 749	24	52	1	11	2	10	181 285
	MIN (*ParkScene*)	22	0	7	19	50	12	12	78 413	12	57	2	17	4	8	230 223	16	56	1	15	4	8	348 898	14	43	2	25	7	10	233 068
		27	0	9	23	42	11	15	62 100	12	63	1	12	3	9	203 785	17	62	1	10	2	8	309 240	15	51	2	17	4	10	189 049
		32	0	10	25	41	9	16	56 387	13	67	1	9	2	9	190 451	17	65	1	8	2	8	288 033	15	57	1	13	3	10	167 093
		37	0	11	29	34	8	18	48 094	13	69	1	8	2	9	182 630	17	68	0	6	1	8	273 078	15	61	1	10	2	11	154 612
720p	MAX (*KristenAndSara*)	22	0	10	25	41	10	15	25 963	–	–	-	–	–	–	–	17	63	1	10	2	8	122 042	13	54	1	17	4	11	73071
		27	0	11	28	36	9	17	22 952	–	–	–	–	–	–	–	17	67	0	7	2	8	113 911	13	60	1	13	3	11	64 926
		32	0	12	29	33	8	19	21 660	–	–	–	–	–	–	–	16	68	0	6	1	8	109 726	12	63	0	10	2	12	60 846
		37	0	12	31	31	7	19	20 347	-	–	–	–	–	–	–	16	69	0	5	1	8	106 673	11	64	0	10	2	12	59 129
	MIN (*Johnny*)	22	0	10	25	40	10	15	25 558	–	–	–	–	–	–	–	15	65	1	9	2	8	115 291	10	57	1	17	4	12	67 801
		27	0	11	28	36	9	17	22 713	–	–	–	–	–	–	–	15	68	0	6	1	9	107 668	10	63	0	12	3	12	59 845
		32	0	12	29	33	8	18	21 152	–	–	-	–	–	–	–	15	69	0	6	1	8	104 725	9	65	0	10	2	13	57 151
		37	0	13	31	30	7	19	20 047	–	–	–	–	–	–	–	15	70	0	5	1	8	102 236	9	67	0	10	2	12	55 665
WVGA	MAX (*RaceHorses*)	22	0	8	19	47	14	13	15 042	20	41	2	21	7	7	63 238	25	42	2	18	7	7	91 583	27	29	3	25	9	8	69 509
		27	0	8	22	43	12	15	12 915	21	48	2	17	5	8	54 141	26	47	1	14	4	7	79 336	29	34	2	20	6	9	57 662
		32	0	10	25	40	10	16	11 308	21	54	1	13	3	8	47 416	26	53	1	11	3	7	70 125	29	40	2	17	4	8	47 767
		37	0	11	27	37	8	17	10 111	20	59	1	10	2	8	43 266	25	57	1	8	2	7	63 874	28	46	1	13	3	9	41157
	MIN (*BQMall*)	22	0	8	20	47	13	13	14 383	14	53	2	18	5	8	47 424	18	52	1	16	5	7	67 698	16	39	2	25	8	9	46 513
		27	0	9	24	42	11	15	12 200	15	59	1	13	4	9	42 263	19	58	1	12	3	8	60 140	17	46	2	20	5	10	38 995
		32	0	10	25	40	10	15	11 072	15	63	1	11	3	9	39 333	19	62	1	9	2	8	55 600	17	52	1	16	4	10	33 769
		37	0	11	28	36	8	17	9983	14	65	1	9	2	9	37 634	18	64	1	7	2	8	52 752	17	56	1	13	3	11	30 864
WQVGA	MAX (*RaceHorses*)	22	0	7	19	48	15	12	3859	17	43	2	22	8	8	14 812	22	42	2	19	7	7	21 405	23	29	3	26	10	8	16 239
		27	0	7	20	48	12	13	3550	18	48	2	18	5	9	12 825	24	46	2	16	5	7	18 918	26	34	2	22	7	8	13 427
		32	0	9	24	41	10	16	2900	18	55	1	14	3	8	11 158	24	53	1	12	3	7	16 578	26	40	2	19	5	9	11 302
		37	0	10	27	38	8	17	2491	18	59	1	10	3	8	10093	24	57	1	9	2	8	14 948	25	46	1	15	3	10	9638
	MIN (*BQSquare*)	22	0	6	17	50	17	11	4320	9	51	2	23	7	8	12 633	10	50	2	22	8	8	18 039	6	37	3	32	13	9	12 900
		27	0	7	19	48	15	12	3759	10	61	1	15	4	8	10 317	12	60	1	15	4	8	14 853	7	47	2	26	8	10	9752
		32	0	8	22	43	13	14	3235	11	67	1	10	3	9	9411	13	66	1	10	3	8	13 200	8	57	1	19	4	11	7823
		37	0	9	25	40	11	16	2840	11	70	0	8	2	9	9037	13	70	0	7	2	8	12 420	8	64	0	13	3	11	6895

a single stage. Preprocessing, memory, and postprocessing functions not belonging directly to any of these coding stages are allocated to *miscellaneous* (*Misc*) group. In addition, Misc group includes coding stages (such as LF) whose relative share is under 1% of the total encoding time.

The overall average shares of these reported encoding stages are gathered in Table X. The AI condition has the lowest complexity since it operates without inter prediction (IME and FME). The inclusion of inter prediction increments the complexities of the RA, LB, and LP conditions approximately by 3.6×, 5.3×, and 3.4× over the AI case, respectively. IME, FME, and MD together contribute over 2/3 of the whole encoding time in the RA and LP cases. The respective share is 3/4 in the LB case. Hence, their acceleration is in the highest priority. Especially, the parameterization of IME has a huge impact on the overall encoding complexity. For example, replacing EPZS with exhaustive full search algorithm would make IME the most complex stage.

QP value also has an impact on the overall encoding time. Incrementing QP value from 22 to 27 reduces the average encoding time by around 15%. The respective decrements are 10% and 8% when QP value is incremented from 27 to 32 and from 32 to 37. All in all, the average cycle count decreases around 29% when changing QP value from 22 to 37.

The dominating roles of the IME, FME, and MD stages in encoding give reason to identify their internal functions

TABLE IX

COMPLEXITY DISTRIBUTION OF HM MP DECODING: THE WORST-CASE AND THE BEST-CASE TEST SEQUENCES

Format	Complexity (Sequence)	QP	AI ED (%)	AI IQ/IT (%)	AI IP (%)	AI MC (%)	AI LF (%)	AI Misc (%)	AI Total (Mcpf)	RA ED (%)	RA IQ/IT (%)	RA IP (%)	RA MC (%)	RA LF (%)	RA Misc (%)	RA Total (Mcpf)	LB ED (%)	LB IQ/IT (%)	LB IP (%)	LB MC (%)	LB LF (%)	LB Misc (%)	LB Total (Mcpf)	LP ED (%)	LP IQ/IT (%)	LP IP (%)	LP MC (%)	LP LF (%)	LP Misc (%)	LP Total (Mcpf)
		22	18	18	22	0	12	29	1824	10	9	7	32	16	27	1196	–	–	–	–	–	–	–			–	–	–	–	–
	MAX	27	13	20	26	0	15	26	1550	6	7	6	36	20	25	971	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*PeopleOnStreet*)	32	10	22	29	0	17	23	1363	4	5	5	41	22	23	840	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		37	7	25	29	0	17	22	1209	2	5	4	47	20	21	751	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1600p		22	18	18	23	0	12	29	1825	6	6	3	51	12	23	841	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	MIN	27	13	22	25	0	15	25	1497	3	4	3	58	12	21	680	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	(*Traffic*)	32	10	25	26	0	16	23	1278	2	4	3	59	12	20	593	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		37	7	30	26	0	16	21	1103	1	4	3	60	13	20	533	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		22	24	14	20	0	8	33	1170	12	8	3	41	12	24	642	14	9	1	41	10	25	742	17	11	3	24	13	33	698
	MAX	27	16	20	24	0	12	27	834	4	5	2	57	12	19	383	4	7	1	57	11	20	396	6	8	1	39	17	29	330
	(*BQTerrace*)	32	11	24	27	0	15	23	687	2	4	2	65	11	17	326	2	5	1	64	10	19	313	2	6	1	49	14	28	235
		37	7	28	28	0	16	22	595	1	3	1	68	11	17	312	1	3	1	68	9	19	278	1	4	1	55	11	27	205
1080p		22	21	18	22	0	10	30	1003	9	10	5	36	15	25	449	10	13	2	35	14	26	475	12	14	4	22	16	32	443
	MIN	27	13	22	25	0	14	25	750	4	9	4	43	16	24	315	4	14	2	39	14	26	309	5	15	3	28	17	31	272
	(*Cactus*)	32	9	26	26	0	16	23	634	2	9	4	46	16	23	271	3	13	2	41	14	27	251	3	15	3	31	16	32	221
		37	6	31	27	0	15	21	549	1	8	3	48	16	23	249	2	12	2	43	14	28	215	2	13	3	33	15	33	189
		22	14	20	26	0	13	26	356	–	–	–	–	–	–	–	5	10	2	34	16	33	100	6	11	2	25	18	38	91
	MAX	27	11	23	27	0	15	24	310	–	–	–	–	–	–	–	3	9	2	34	16	37	75	3	10	2	26	17	42	69
	(*FourPeople*)	32	8	27	27	0	16	23	275	–	–	–	–	–	–	–	2	8	2	32	16	40	64	2	9	2	27	16	44	60
		37	6	31	27	0	16	21	245	–	–	–	–	–	–	–	1	8	2	30	16	43	56	1	8	2	26	15	47	54
720p		22	12	28	23	0	13	25	289	–	–	–	–	–	–	–	3	8	1	46	13	28	103	5	10	1	32	17	35	91
	MIN	27	8	33	24	0	13	22	254	–	–	–	–	–	–	–	1	6	1	47	12	32	79	2	8	2	35	15	39	67
	(*Johnny*)	32	6	36	25	0	14	20	232	–	–	–	–	–	–	–	1	6	1	44	13	35	67	1	6	2	35	14	42	59
		37	4	39	25	0	14	19	215	–	–	–	–	–	–	–	1	5	2	39	13	40	58	1	5	2	34	14	44	54
		22	25	12	20	0	7	36	301	12	10	4	39	9	27	121	13	11	2	37	9	28	142	15	12	3	25	11	33	131
	MAX	27	21	13	24	0	10	32	242	7	8	4	46	10	24	94	8	10	2	44	11	25	104	9	11	3	32	13	31	91
	(*PartyScene*)	32	17	15	27	0	13	29	197	4	7	4	53	11	22	79	5	8	2	50	11	24	81	6	10	3	39	13	29	68
		37	12	17	30	0	15	25	161	2	6	3	58	10	20	70	3	8	2	54	10	23	65	3	10	3	44	12	28	55
WVGA		22	19	16	24	0	11	30	199	8	8	4	43	11	26	87	8	11	2	40	12	27	96	10	13	3	27	15	32	86
	MIN	27	15	19	27	0	13	27	166	5	8	4	48	12	24	72	5	10	2	44	13	26	75	6	12	3	32	15	31	65
	(*BQMall*)	32	11	22	28	0	15	24	141	3	7	4	52	12	23	63	3	9	2	47	12	26	62	4	11	3	36	14	31	53
		37	8	26	28	0	15	22	121	2	7	3	54	12	22	58	2	9	2	49	11	27	53	2	10	3	40	13	32	45
		22	22	17	20	0	8	33	60	11	11	6	33	10	29	37	12	13	4	32	10	29	40	13	14	5	24	12	32	37
	MAX	27	18	19	23	0	11	29	49	7	11	6	37	12	27	30	8	13	4	36	12	27	31	8	15	4	29	14	30	28
	(*RaceHorses*)	32	13	22	26	0	14	25	41	4	12	5	42	13	24	25	5	14	3	39	12	26	25	5	15	4	33	14	28	23
		37	9	26	28	0	14	22	34	3	12	4	46	12	22	22	3	15	3	42	11	26	20	3	16	4	37	12	28	19
WQVGA		22	20	19	22	0	9	30	52	10	13	7	29	11	29	30	12	15	4	29	11	30	32	13	17	6	20	12	33	29
	MIN	27	15	23	24	0	11	26	43	7	14	6	34	12	27	25	8	16	4	32	12	28	25	9	17	6	24	13	31	23
	(*BasketballPass*)	32	11	26	26	0	13	23	36	4	14	6	39	13	25	21	5	16	4	35	12	27	21	5	18	5	28	14	30	19
		37	8	30	27	0	14	21	31	2	14	5	43	12	24	19	3	17	3	38	11	27	18	3	19	4	31	13	30	16

TABLE X

AVERAGE SHARES OF THE MOST COMPLEX ENCODING STAGES

Encoding stage	AI	RA	LB	LP
IME	0%	16%	18%	17%
FME/MD	9%	55%	59%	49%
IP	24%	1%	1%	1%
T/Q/IQ/IT	41%	14%	11%	18%
EC	11%	4%	3%	5%
Misc.	15%	10%	8%	10%

TABLE XI

AVERAGE SHARES OF THE MOST COMPLEX ENCODING FUNCTIONS

Function	AI	RA	LB	LP
Interpolation	0%	37%	38%	31%
SATD computation	9%	16%	18%	15%
SAD computation	0%	10%	12%	11%
Sum	9%	63%	68%	57%

more accurately. The most complex functions among these stages are IPOL in the FME stage, SATD computation in the FME/MD stages, and SAD computation in the IME stage. Their average complexities under the AI, RA, LB, and LP configurations are tabulated in Table XI. In the AI case, the shares of IPOL, SATD, and SAD are limited to SATD computation in MD. In the other conditions, these functions take the major part of the whole encoding complexity (57%–68%). On average, IPOL and SATD contribute about 95% of the FME/MD complexity, whereas SAD computation is responsible for around 65% of the IME complexity.

Table XII reports the approximated operation counts of these IPOL, SATD, and SAD functions when the worst case 1080p sequence (*BasketballDrive*) of our test set (Table VIII) is encoded at QP_{HM} = 22. The operation counts are tabulated as *Giga operations per second* (*GOPS*) required for real-time (50 fps) encoding in the AI, RA, LB, and LP cases.

The analysis covers the arithmetic (addition, subtraction, multiplication, absolute value, and comparison) and memory operations (load and store) that are needed to implement the fundamental algorithms of these functions. The excluded operations include HM-specific control and logic operations whose share of the overall complexity is only marginal. The reported operation counts have been gathered from the platform-independent C++ source code of HM 6.0. Hence, they are only approximations of the actual platform-specific operation counts that are strongly dependent on the underlying hardware platform and compiler.

The reported results have been allocated to main subfunctions of IPOL, SAD, and SATD. IPOL subfunctions include 4-tap and 8-tap filters whereas SATD and SAD subfunctions are dedicated to different PU sizes. In IPOL and SATD functions, the *Other* groups contain operations not belonging directly to any of their main subfunctions.

The computation load of all these functions is almost entirely originated from the basic arithmetic operations. Hence, they are all well suited to hardware acceleration. However, the number of memory operations is close to that of arithmetic operations, so meeting the high memory bandwidth demands may easily play the most critical role in hardware implementations.

TABLE XII
OPERATION COUNTS OF THE MOST COMPLEX ENCODING FUNCTIONS (*BasketballDrive* AT QP_{HM} = 22)

Condition	Oper.	Function													
		IPOL				SATD				SAD					
		4-tap (GOPS)	8-tap (GOPS)	Other (GOPS)	Total (GOPS)	4 × 4 (GOPS)	8 × 8 (GOPS)	Other (GOPS)	Total (GOPS)	4 × x (GOPS)	8 × x (GOPS)	16 × x (GOPS)	32 × x (GOPS)	64 × x (GOPS)	Total (GOPS)
AI	add	0.0	0.0	0.0	0.0	13.6	62.2	0.0	75.8	0.0	0.0	0.0	0.0	0.0	0.0
	sub	0.0	0.0	0.0	0.0	10.9	56.9	0.0	67.8	0.0	0.0	0.0	0.0	0.0	0.0
	mul	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	abs	0.0	0.0	0.0	0.0	3.6	14.2	0.0	17.8	0.0	0.0	0.0	0.0	0.0	0.0
	cmp	0.0	0.0	0.0	0.0	0.9	5.3	0.0	6.2	0.0	0.0	0.0	0.0	0.0	0.0
	load	0.0	0.0	0.0	0.0	36.3	213.3	0.0	249.6	0.0	0.0	0.0	0.0	0.0	0.0
	store	0.0	0.0	0.0	0.0	18.1	99.5	0.0	117.7	0.0	0.0	0.0	0.0	0.0	0.0
RA	add	56.8	1065.5	45.2	1167.5	80.7	456.2	19.2	556.1	95.0	226.9	195.9	146.4	94.0	758.2
	sub	0.0	0.0	0.0	0.0	64.5	417.1	0.0	481.7	79.1	204.2	185.0	142.1	92.6	703.1
	mul	50.8	1033.9	0.0	1084.7	0.0	0.0	0.0	0.0	4.0	7.6	2.4	0.6	0.1	14.7
	abs	0.0	0.0	0.0	0.0	21.5	104.3	0.0	125.8	63.3	181.5	174.2	137.8	91.1	647.9
	cmp	14.7	139.8	39.7	194.2	5.4	39.1	10.9	55.4	15.8	22.7	10.9	4.3	1.4	55.1
	load	103.0	2075.3	36.0	2214.2	215.1	1564.2	7.4	1786.7	138.5	385.9	355.4	277.6	182.7	1340.0
	store	14.1	136.7	36.0	186.8	107.5	730.0	0.0	837.5	2.0	3.8	1.2	0.3	0.1	7.4
LB	add	62.1	1557.8	68.7	1688.6	120.3	684.4	28.7	833.4	176.1	416.8	365.0	277.6	179.4	1414.8
	sub	0.0	0.0	0.0	0.0	96.3	625.7	0.0	722.0	146.7	375.1	344.7	269.4	176.7	1312.7
	mul	55.4	1511.4	0.0	1566.9	0.0	0.0	0.0	0.0	7.3	14.1	4.4	1.2	0.3	27.2
	abs	0.0	0.0	0.0	0.0	32.1	156.4	0.0	188.5	117.4	333.5	324.4	261.2	173.9	1210.5
	cmp	16.1	204.4	224.0	444.4	8.0	58.7	16.3	83.0	29.3	41.7	20.3	8.2	2.7	102.2
	load	112.4	3033.8	202.8	3349.0	320.9	2346.4	11.1	2678.3	256.8	709.1	662.0	526.0	348.7	2502.5
	store	15.4	199.9	202.8	418.1	160.4	1095.0	0.0	1255.4	3.7	7.0	2.2	0.6	0.1	13.6
LP	add	24.0	756.3	38.8	819.0	64.8	368.6	15.4	448.8	123.8	307.2	283.7	218.5	141.3	1074.5
	sub	0.0	0.0	0.0	0.0	51.9	337.0	0.0	388.8	103.2	276.5	267.9	212.0	139.2	998.8
	mul	21.4	733.7	0.0	755.2	0.0	0.0	0.0	0.0	5.2	10.2	3.4	0.9	0.2	19.9
	abs	0.0	0.0	0.0	0.0	17.3	84.2	0.0	101.5	82.6	245.7	252.1	205.6	137.0	923.1
	cmp	6.2	99.2	107.1	212.5	4.3	31.6	8.7	44.6	20.6	30.7	15.8	6.4	2.1	75.7
	load	43.4	1472.8	97.1	1613.3	172.9	1263.6	5.7	1442.2	180.6	522.2	514.5	414.0	274.7	1906.0
	store	5.9	97.0	97.1	200.0	86.4	589.7	0.0	676.1	2.6	5.1	1.7	0.5	0.1	10.0

TABLE XIII
AVERAGE SHARES OF THE MOST COMPLEX DECODING STAGES

Decoding stage	AI	RA	LB	LP
LF	13%	13%	12%	14%
MC	0%	47%	44%	34%
IP	25%	4%	2%	3%
IQ/IT	23%	9%	11%	12%
ED	13%	5%	5%	6%
Misc.	26%	22%	26%	31%

B. Complexity Analysis of HM MP Decoder

The most complex stages of HM decoder are ED, IQ/IT, IP, MC, and LF (Table IX). The overall average shares of these stages are summarized in Table XIII. As in the encoder analysis, the remaining functions are allocated to Misc group.

The AI configuration has to cope with the highest bit rate due to which it also has the highest complexity in decoding. The decoding complexities of the RA, LB, and LP configurations are approximately halved from that of the AI case. In RA, LB, and LP conditions, MC is the most complex stage. The complexity distribution in the RA condition corresponds to our previous experiments on HM LC (HM 3.0) [27] with an average deviation of ±2 pps per individual share. As in encoding, QP value also impacts on overall decoding time. Incrementing QP value from 22 to 27 reduces the average decoding time by around 23%. The decrements are 17% and 13% when QP value is incremented from 27 to 32 and from 32 to 37, respectively. On average, the cycle count decreases around 44% between QP values of 22 and 37.

Accelerating the most complex functions such as MC is recommended in decoding, but an adequate decoding performance is typically obtainable through processor-based acceleration. However, HEVC codec is strongly asymmetrical in terms of complexity, so sufficient encoding performance tends to be out of reach unless the most complex encoding functions are off-loaded to special hardware accelerators.

C. Encoder/Decoder Complexity Comparison

Table XIV tabulates the minimum, maximum, and average complexity ratios of HM encoder and decoder under the AI, RA, LB, and LP conditions. The average complexity ratio of the entire test set is around 500, but it varies between 2 and 3 orders of magnitude (62–1469). According to our analysis, the complexity ratio follows the share of the inter prediction.

D. Comparison Between HM MP and JM HiP Codecs

Fig. 5 depicts the average QP-specific complexities of HM MP and JM HiP encoders in terms of Mcpf at 1080p and WQVGA resolutions. The bar diagrams indicate the overall

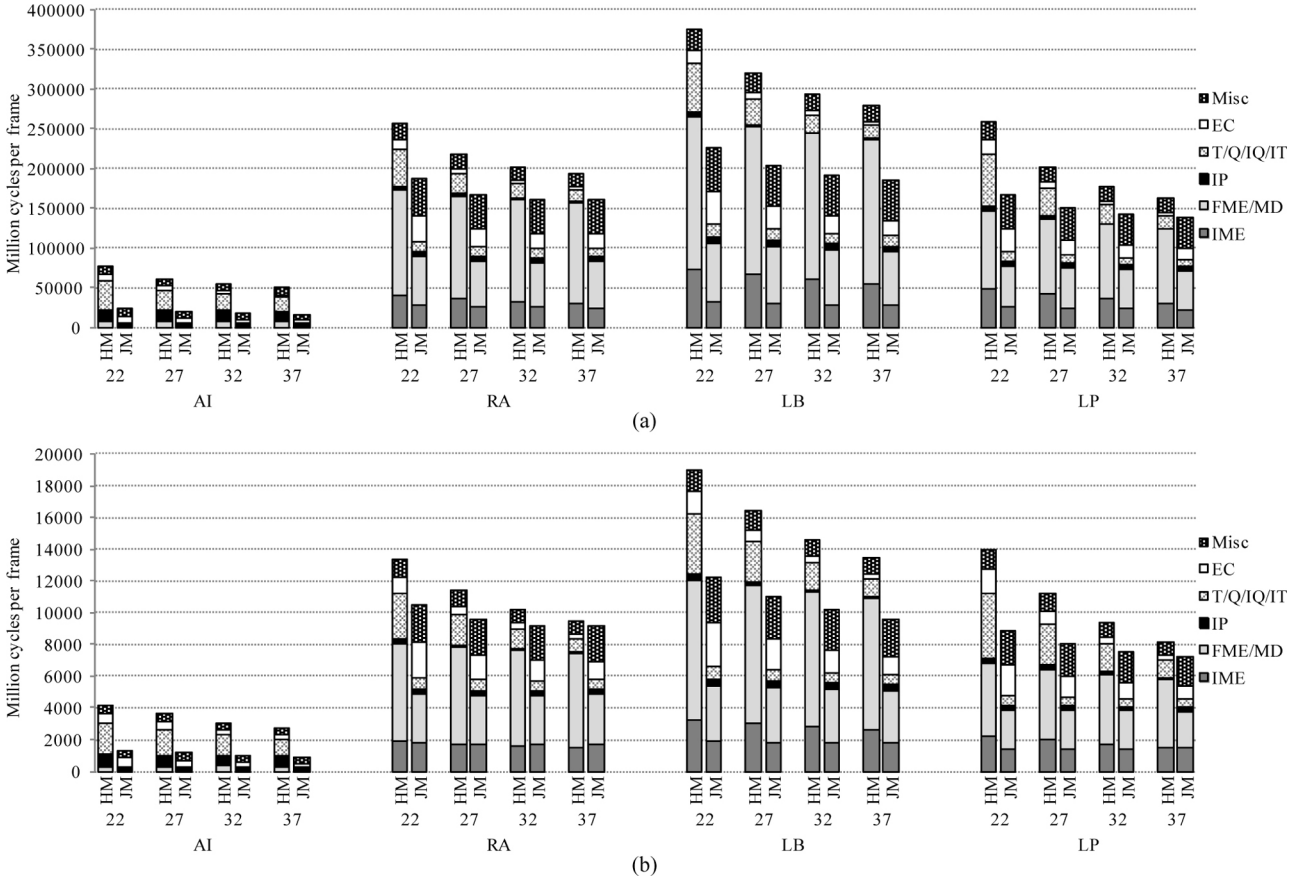

Fig. 5. Average complexity comparison between HM MP and JM HiP encoders. (a) 1080p sequences. (b) WQVGA sequences.

TABLE XIV
COMPLEXITY RATIO OF HM MP ENCODER AND DECODER

HM encoder versus HM decoder	AI	RA	LB	LP
Min complexity ratio	62	381	571	426
Max complexity ratio	98	651	1469	900
Avg. complexity ratio	77	501	851	608

encoding complexities as well as the portions of the individual coding stages. The corresponding bar diagrams for HM MP and JM HiP decoders are illustrated in Fig. 6. In both cases, QP_{HM} values are 22, 27, 32, and 37, whereas the respective QP_{JM} values have been obtained from Table IV. For brevity, a detailed complexity comparison between HM MP and JM HiP is limited to these two resolutions.

For the complete test set (the Classes *A*–*E*), the average complexity ratios of HM and JM encoders are 3.2× in the AI case, 1.2× in the RA case, 1.5× in the LB case, and 1.3× in the LP case. These complexity ratios do not change as a function of resolution but most of them decrease when QP value increases. When QP_{HM} = 22, the respective complexity ratios of the RA, LB, and LP configurations are 1.3×, 1.6×, and 1.5× and they decrease down to 1.1×, 1.5×, and 1.2× when QP_{HM} value is incremented from 22 to 37. In the AI configuration, the ratio remains the same with different QP values.

The complexity ratios of HM and JM decoders are 2.0×, 1.6×, 1.5×, and 1.4× in the AI, RA, LB, and LP configurations, respectively. Incrementing QP_{HM} value from 22 to 37 increases complexity ratio by 10% in the AI case but decreases the ratio by 11% in the LB case. In the other cases, the complexity ratio remains the same.

E. Considerations on Real-Time HEVC Video Codecs

HM MP and JM HiP codecs are well known, publicly available, and platform-independent implementations that incorporate practically all normative and nonnormative parts of HEVC MP and AVC HiP. Hence, they are the best references for the fair RDC comparison between HEVC MP and AVC HiP codecs. However, HM and JM are targeted for research and conformance testing rather than practical real-time codecs which have to meet practical limitations in execution speed, chip size, and power consumption.

The real-time HEVC decoding has already been addressed by proprietary HM 4.0-based decoders [24]–[26] optimized for mobile ARM and stationary x86 platforms. In the RA case, the optimized HEVC HE decoder in [25] is able to decode WVGA format at 30 fps when mapped on a single 1.5 GHz Snapdragon processor core. According to [26], the same decoding speed (WVGA at 30 fps) is achieved with the optimized HEVC LC decoders on a 1 GHz ARM Cortex-A9 core. On a 2.66 GHz Core i5 processor, the optimized HEVC LC decoders are able to decode 1080p format up to 60 fps with a single core [24].

Decoding 1080p resolution at 60 fps with HM would require almost 40 000 Million cycles per second (642 Mcpf) under the

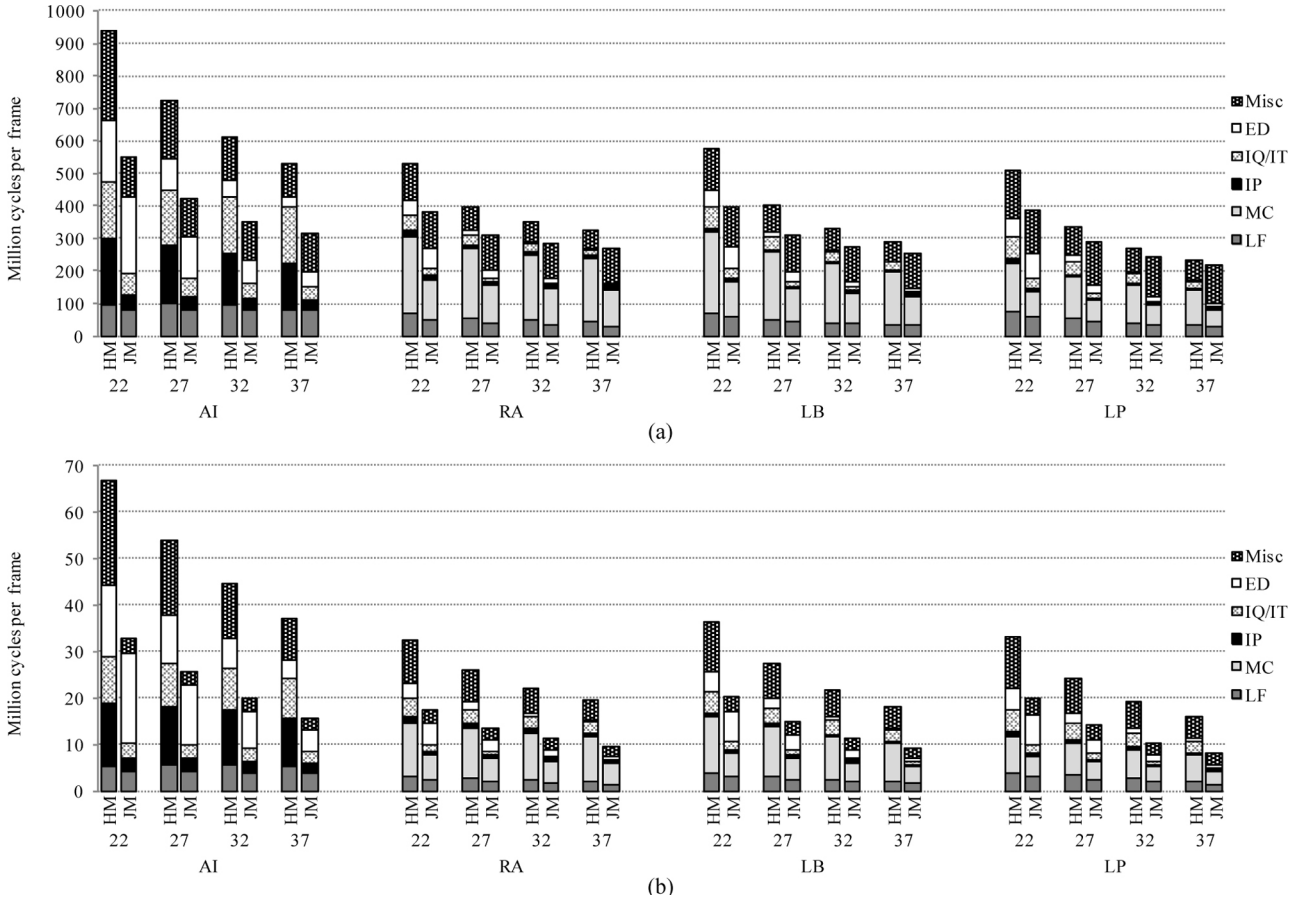

Fig. 6. Average complexity comparison between HM MP and JM HiP decoders. (a) 1080p sequences. (b) WQVGA sequences.

RA configuration in the worst case (see Table IX). In theory, that complexity would be tackled with 15 cores of Core i5 processor clocked at 2.66 GHz, so HM MP complexity can be coarsely estimated to be 15 times that of the optimized HEVC LC decoder in [24]. In practice, the comparison is not so straightforward because the optimized decoders utilize SIMD acceleration (MMX/SSE on x86 and NEON on ARM) for the most complex functions, mapping HM to 15 cores would cause overheads due to nonoptimal scaling, etc. However, the profiling results provided for these optimized HEVC decoders are still quite consistent with ours (see Table IX). For example, the relative shares of the decoding functions reported in [26] correspond to our results with an average deviation of ± 4 pps per individual share. Hence, the profiling results reported in this paper for HM can be seen as valid estimates also for the optimized HEVC software codecs.

Currently, HM is the only publicly known HEVC encoder, but optimized real-time HEVC encoders are expected to be released in the near future. The complexity ratio between HEVC MP and AVC HiP encoders is only a fraction of the respective processing technology development from the announcement of AVC. Hence, by assuming that the relative speed-up of optimized HEVC encoders is analogous to that of optimized AVC encoders, the real-time performance of HEVC encoder is well within the range of the current technology. The encoding speed of HEVC can be also enhanced by excluding nonnormative encoding tools at cost of quality or without quality loss by off-loading the most complex functions to hardware accelerators and/or special-purpose processors. Our recommendation is to start off-loading from the IME, FME, and MD stages in which the hardware-oriented IPOL, SAD, and SATD functions are the most complex ones (see Table XI).

The future trend is that the processing performance will continue to develop faster than transmission and storage technologies [31]. This trend will further promote HEVC because of its capability to almost halve the bit rate. Due to these reasons, we forecast rapid proliferation of HEVC in the next-generation video products and services.

VI. Conclusion

This paper presented the results of the comparative RDC analysis of HEVC MP (HM 6.0) and AVC HiP (JM 18.0) video codecs under the AI, RA, LB, and LP configurations. The resolutions of the test sequences varied from WQVGA up to 4K and the operating points of HM were examined with QP values of 22, 27, 32, and 37. This RDC analysis relied on PSNR as an objective quality measure whereas complexities were obtained through cycle-level profiling with Intel VTune. The fair comparison was attained by configuring JM HiP to conform HM MP settings and coding configurations.

Our main results are gathered in Table XV. On average, HM MP reduces bit rate over JM HiP almost 37% with an equivalent objective quality and at around 1.4× coding

TABLE XV
RDC SUMMARY OF HEVC MP (HM 6.0) AND AVC HIP (JM 18.0)

HM MP versus JM HiP	AI	RA	LB	LP
BD-rate	23%	35%	40%	35%
Encoding complexity	3.2×	1.2×	1.5×	1.3×
Decoding complexity	2.0×	1.6×	1.5×	1.4×

complexity when all essential coding tools of HM MP and JM HiP are used. Furthermore, the coding gain of HM MP is shown to increase as a function of the resolution. These HEVC characteristics are well balanced with the current technology roadmap according to which relative development of processing performance in stationary and mobile terminals is faster than that of transmission and storage technologies.

The reported results reveal the bottlenecks of the HM software codec and the given implementation guidelines can be used to evaluate the requirements of the underlying codec architecture. In general, off-loading the most complex coding algorithms such as ME to dedicated accelerators will be needed particularly in mobile devices to meet practical limitations in execution speed, chip size, and power consumption. HEVC MP as a part of the released HEVC draft standard ensures that the results of this RDC analysis will remain as a valid platform-independent point of reference for future HEVC software codec implementations.

REFERENCES

[1] *Advanced Video Coding for Generic Audiovisual Services,* ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC, Mar. 2009.

[2] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 6,* document JCTVC-H1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, Feb. 2012.

[3] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 8,* document JCTVC-J1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Stockholm, Sweden, Jul. 2012.

[4] F. Bossen, V. Drugeon, E. Francois, J. Jung, S. Kanumuri, M. Narroschke, H. Sasai, J. Sole, Y. Suzuki, T. K. Tan, T. Wedi, S. Wittmann, P. Yin, and Y. Zheng, "Video coding using a simplified block structure and advanced coding techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1667–1675, Dec. 2010.

[5] W. J. Han, J. Min, I. K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M. S. Cheon, N. Shlyakhov, K. McCann, T. Davies, and J. H. Park, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.

[6] M. Karczewicz, P. Chen, R. L. Joshi, X. Wang, W. J. Chien, R. Panchal, Y. Reznik, M. Coban, and I. S. Chong, "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, pp. 1698–1708, Dec. 2010.

[7] D. Marpe, H. Schwarz, S. Bosse, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Sühring, M. Winken, and T. Wiegand, "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, pp. 1676–1687, Dec. 2010.

[8] K. Ugur, K. Andersson, A. Fuldseth, G. Bjøntegaard, L. P. Endresen, J. Lainema, A. Hallapuro, J. Ridge, D. Rusanovskyy, C. Zhang, A. Norkin, C. Priddle, T. Rusert, J. Samuelsson, R. Sjöberg, and Z. Wu, "High performance, low complexity video coding and the emerging HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1688–1697, Dec. 2010.

[9] *Joint Collaborative Team on Video Coding Reference Software, ver. HM 6.0* [Online]. Available: https://hevc.hhi.fraunhofer.de/

[10] K. McCann, B. Bross, I. K. Kim, K. Sugimoto, and W. J. Han, *HM6: High Efficiency Video Coding (HEVC) Test Model 6 Encoder Description,* document JCTVC-H1002, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, Feb. 2012.

[11] F. Bossen, *Common Test Conditions and Software Reference Configurations,* document JCTVC-H1100, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, Feb. 2012.

[12] F. De Simone, L. Goldmann, J. S. Lee, and T. Ebrahimi, "Towards high efficiency video coding: Subjective evaluation of potential coding technologies," *J. Vis. Commun. Image R.*, vol. 22, no. 8, pp. 734–748, Nov. 2011.

[13] *Joint Video Team Reference Software, ver. JM 18.0* [Online]. Available: http://iphome.hhi.de/suehring/tml/.

[14] F. Kossentini, N. Mahdi, H. Guermazi, M. Horowitz, S. Xu, B. Li, G. J. Sullivan, and J. Xu, *Informal Subjective Quality Comparison of Compression Performance of HEVC Working Draft 5 with AVC High Profile,* document JCTVC-H0562, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, Feb. 2012.

[15] Y. Zhao, L. Yu, Z. Chen, and C. Zhu, "Video quality assessment based on measuring perceptual noise from spatial and temporal perspectives," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 12, pp. 1890–1902, Dec. 2011.

[16] Y. Zhao and L. Yu, *Coding Efficiency Comparison Between HM5.0 and JM16.2 Based on PQI, PSNR and SSIM,* document JCTVC-H0063, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, Feb. 2012.

[17] B. Li, G. J. Sullivan, and J. Xu, "Compression performance of high efficiency video coding (HEVC) working draft 4," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 886–889.

[18] B. Li, G. J. Sullivan, and J. Xu, *Comparison of Compression Performance of HEVC Draft 6 with AVC High Profile,* document JCTVC-I0409, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Apr. 2012.

[19] B. Li, G. J. Sullivan, and J. Xu, *Comparison of Compression Performance of HEVC Draft 7 with AVC High Profile,* document JCTVC-J0236, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Stockholm, Sweden, Jul. 2012.

[20] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves,* document VCEG-M33, Austin, TX, Apr. 2001, pp. 1–4.

[21] F. Bossen, D. Flynn, and K. Sühring, *"AHG Report: Software Development and HM Software Technical Evaluation,* document JCTVC-G003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Nov. 2011.

[22] S. Park, J. Park, and B. Jeon, *Report on the Evaluation of HM versus JM,* document JCTVC-D181, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Daegu, Korea, Jan. 2011.

[23] T. Anselmo and D. Alfonso, *HM Decoder Complexity Assessment on ARM,* document JCTVC-G262, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Nov. 2011.

[24] K. McCann, J. Y. Choi, K. Pachauri, K. P. Choi, C. Kim, Y. Park, Y. J. Kwak, S. Jun, M. Choi, H. Yang, and J. Park, *HEVC Software Player Demonstration on Mobile Devices,* document JCTVC-G988, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Nov. 2011.

[25] K. Veera, R. Ganguly, B. Zhou, N. Kamath, S. Chowdary, J. Du, I. S. Chong, and M. Coban, *A Real-Time ARM HEVC Decoder Implementation,* document JCTVC-H0693, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, Feb. 2012.

[26] F. Bossen, *On Software Complexity,* document JCTVC-G757, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Nov. 2011.

[27] M. Viitanen, J. Vanne, T. D. Hämäläinen, M. Gabbouj, and J. Lainema, "Complexity analysis of next-generation HEVC decoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 882–885.

[28] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 934–947, Oct. 2002.

[29] S. Sun and J. Reichel, *AHG Report on Spatial Scalability Resampling,* document JVT-R006, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Bangkok, Thailand, Jan. 2006.

[30] B. Li, G. J. Sullivan, and J. Xu, *RDO with Weighted Distortion in HEVC,* document JCTVC-G401, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Nov. 2011.

[31] International Technology Roadmap for Semiconductors (ITRS). (2011). *Edition: System Drivers* [Online]. Available: http://public.itrs.net

Jarno Vanne (M'02) received the M.Sc. degree in information technology and the Ph.D. degree in computing and electrical engineering from the Tampere University of Technology (TUT), Tampere, Finland, in 2002 and 2011, respectively.

He is currently a Research Fellow with the Department of Computer Systems, TUT. His current research interests include video and image processing systems, video coding standards, motion estimation, parallel memories, and computer arithmetic.

Marko Viitanen (M'09) is currently pursuing the M.Sc. degree in computer systems with the Tampere University of Technology (TUT), Tampere, Finland.

He is currently a Research Assistant with the Department of Computer Systems, TUT. His current research interests include video compression, video coding standards, and performance analysis.

Timo D. Hämäläinen (M'95) received the M.Sc. and Ph.D. degrees from the Tampere University of Technology (TUT), Tampere, Finland, in 1993 and 1997, respectively.

He has been a Full Professor with the Department of Computer Systems, TUT, since 2001. He is an author of more than 60 journals and 200 conference publications. He holds several patents. His current research interests include design methods and tools for multiprocessor systems-on-a-chip and parallel video codec implementations.

Antti Hallapuro received the M.Sc. degree in computer science from the Tampere University of Technology, Tampere, Finland, in 2010.

He joined the Nokia Recearch Center, Tampere, in 1998, where he has been engaged in video coding related topics. He has participated in AVC and HEVC video codec standardizations and is an author and co-author of several input documents and related academic papers. He has contributed to productization of high performance AVC codecs for various computing platforms. He is currently working on topics of the next generation video coded standardization. His current research interests include practical video coding and processing algorithms and optimized implementations.