Performance Evaluation of H.265/MPEG-HEVC Encoders for 4K Video Sequences

Qingxiong Huangyuan[* †], Li Song[* †], Zhengyi Luo[† ◊], Xiangwen Wang [† ◊], Yanan Zhao[* †]
[*]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China
[†]Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai, China
[◊]School of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai, China
{hyqx, song_li, xkyang@sjtu.edu.cn, lzy@shiep.edu.cn, wxw21st@gmail.com, ya_nanzhao@hotmail.com}

Abstract— **The H.265/MPEG-HEVC is the latest video coding standard, which achieves an increase of about 50% in coding efficiency compared to its predecessor H.264/MPEG-AVC. Ever since H.265/MPEG-HEVC was designed to replace almost all existing H.264/ MPEG-AVC codecs, high-resolution video coding beyond High Definition (4K, 8K, etc.) has drawn more attention. On the other hand, it's well known that reference implementation of HEVC codec, HM, acts an important role during standardization, particularly for evaluation of rate distortion performance of different tools. However, HM is far from a practical codec because of very slow coding speed even on modern multi-core computers. Up to now except for HM few comparisons are known about both the coding performance and the coding speed of practical HEVC encoders for high resolution video sequences. To address this issue, this paper conducts a comprehensive evaluation of latest high performance H.265/MPEG-HEVC encoders, including the open source encoder—x265 and the commercial encoder—DivX265, based on default parameters and a new open 4K video database. Furthermore, latest HM and x264 are also included for performance anchors. The experimental results show DivX265 provides average bit-rate savings of 4.79% relative to HM while x265 with default preset achieves an average reduction of 3.21% in terms of BD-BR saving. In addition, different presets of x265 make a good tradeoff between coding speed and R-D performance while DivX265 is almost as fast as x265 ultrafast preset. We believe such evaluation information could provide a more comprehensive picture of state-of-the-art H.265/MPEG-HEVC encoders.**

I. INTRODUCTION

In the past decade, people's demands on higher resolution videos and better visual experience promote the great advances in coding field. With the increased bandwidth and the improved adaptability to the changes of the managed network, the HD videos have been widely used in many occasions and the UHD videos have also walked into our life. The new coding standard—H.265/MPEG-HEVC emerges as the times require and has gained increasing popularity in various applications.

Video coding standards evolved through the development of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Pictures Expert Group (MPEG). The two organizations developed the successful H.262/MPEG-2 Video [2] and H.264/MPEG-4 Advanced Video Coding (AVC) [3] standards, which have been widely accepted and adopted. Then they cooperated with each other and formed a group known as Joint Collaborative Team on Video Coding (JCT-VC) [4] to establish the next generation standard of video coding, the High-Efficiency Video Coding (HEVC) standard, which was officially finalized in January 2013. The newly designed standard is expected to suit all H.264/MPEG-AVC applications and performs well in different scenarios, including low and high bit rates, low and high resolutions, broadcast, DVD storage and RTP/IP packet networks [5].

The H.265/MPEG-HEVC standard aims to double the compression ratio of H.264/AVC High profile with comparable image quality [6]. It adopts many new tools which represent the latest achievements in the field of video coding and finally achieves a significant compression ratio improvement compared with H.264/MPEG-AVC. H.265/MPEG-HEVC adopts an enhanced hybrid spatial-temporal prediction model [7], which uses flexible partitioning and introduces Coding Tree Units but not Macro Block (MB) used in H.264. Besides, in intra coding of HEVC, 35 directional modes are used for prediction. To accelerate coding, H.265/MPEG-HEVC has a superior parallel processing architecture for speed improvement on multi-core platforms.

Many previous works have been done to evaluate the R-D performance of H.265/MPEG-HEVC and H.264/MPEG-AVC. Reviewing these recent related works may help us conduct our evaluation in different dimensions with additional information. Vivienne et al. [8] conducted a comparison between HEVC and H.264/MPEG-AVC in terms of CABAC entropy coding improvements and analyzed the throughput of HEVC vs AVC using B-D cycle measurements. Zhou et al. [9] compared the lossless coding and lossy coding of HEVC and proposed a new SAP method to further improve the coding efficiency of HEVC lossless mode. Ohm et al. [10] did a comparison on WVGA and HD videos and analyzed the R-D performance from H.262 to HEVC, where special emphasis was given to the various settings and tools of HEVC that are relevant to coding efficiency. Nguyen et al. [11] evaluated the intra coding of HEVC and H.264/AVC together with popular image coding standards like JPEG and JPEG 2000, and presented average bit-rate savings for HEVC and H.264/AVC intra coding. Grois et al. [12] presented a performance comparison of H.265/MPEG-HEVC, VP9 and H.264/MPEG-

AVC encoders, and showed VP9 was inferior to both H.264/MPEG-AVC and H.265/MPEG-HEVC. Vanne et al. [13] analyzed rate distortion and complexity of the HEVC reference video encoder (HM), and compared the results with AVC reference encoder (JM) under the all-intra (AI), random access (RA), low delay B (LB) and low delay P (LP) coding configurations.

All the works related to Rate-Distortion-Complexity performance evaluation can be classified into two categories: (1) The comparison was performed between different encoders under the same coding standard (2) The comparison was conducted between HEVC and other coding standards like H.264/AVC using different coding modes. Note that almost all the papers and research are based on the reference encoders, e.g. HM and JM, which are only suitable for theoretical research. However, as many optimizations have been done since HEVC standard was released in 2013, people urgently need to know about performance (both compression and speed) of practical encoders at present. Moreover, many of these comparisons focus on relatively low resolution and seldom select UHD videos as their test video sequences. Even if some test sequences like class A used for HEVC are extracted from raw full 4K sources, they are only sub-pictures with smaller size (2560x1600).

To address the above issues, this paper conducts a comprehensive comparison of modern H.265/MPEG-HEVC encoders for full 4K video sequences. We choose two practical encoders—x265[14] and DivX265[15], and compare them based on a 4K video database[16], which is open for public research and were captured by the masterpiece of Sony-F65. More information can be found at [17].

In practical applications both high compression and fast coding are desired for 4K video coding. In this paper, four presets of x265 are chosen to get full information about x265 in different respects. HM and x264 are also selected as the possible upper bound and the lower bound of coding performance. We compare the R-D curves and speed of each encoder and analyze the performance in detail. It should be noted that as not all encoding parameters, e.g. motion vector range of DivX265, can be specified, and the default parameters of HM may be quite different from those of x265 and DivX265, the evaluation can only draw a rough picture of current H.265/HEVC encoders. But we believe the rate-distortion-complexity evaluation can still provide a helpful reference for practical applications.

II. EVALUATION METHODOLOGY

The 4K test sequences in this paper are all in raw YUV 4:2:0 formats. We choose these sequences because of their high resolution and diverse signal characteristics. All sequences have sufficient frames and the video contents of different sequences are quite different. We believe these sequences can provide us a convincing result for the comparison between different encoders.

A. Encoder Settings

In this paper, different presets of x265, HM, x264[18] and DivX265 are chosen for a fair comparison. In particular, the publicly available HEVC encoder－x265, and the private but open HEVC encoder－DivX265 are tested with appropriate presets or parameters under the same platform (Intel Xeon CPU E5-2660 @ 2.20GHz, 16 cores, 24GB physical memory).

The latest version of DivX265 is v1.2.24, which has improved both encoding speed and efficiency dramatically. It provides four different encoding modes (1 - fastest, 2 - balanced (default), 3 - higher quality (experimental), 4-highest quality (experimental)) and wave-front parallel processing is employed to accelerate its coding speed. DivX265 is among the first to deliver tools for HEVC contents and in our work we choose the *default preset* for the comparison in light of its stability and good tradeoff between performance and speed.

TABLE I
PRESETS AND KEY PARAMETERS FOR X265

Preset	Key Parameters
medium (default)	rc-lookahead=20 ctu = 64 bframes = 4 ref = 3 merange = 57 sao = 1
faster	rc-lookahead = 15 ctu = 64 bframes = 4 ref = 3 merange = 57
superfast	rc-lookahead = 10 ctu = 32 bframes = 4 ref = 1 merange = 44
ultrafast	rc-lookahead = 10 ctu = 32 bframes = 4 ref = 1 merange = 25

x265 is an open source implementation of HEVC encoding and targets at real-time coding on generic multicore CPU based platforms. It inherits many key modules from HM, especially those under CTU levels, to maintain high compression performance. At the same time, extensive platform level speeding-up, e.g. assembly optimization and multi-threaded parallel acceleration, has also been done. As a result, the encoding speed of x265 is almost hundreds of times of original HM on a modern multicore computer. Now x265 has been used by some famous tools like FFMPEG and VLC. Like x264, x265 also offers several presets with different kinds of encoder features such as weighted prediction for P slices, adaptive B-frame placement and all partitions in inter P. In this paper, the presets slower than default (*slow*, *slower*, *very slow* and *placebo*) are excluded from comparison and another four presets of x265 are selected. Table I gives some key parameters for each preset. Note that the default preset generally has a good tradeoff between compression performance and coding speed while other presets are preferred by fast encoding applications.

TABLE II
KEY PARAMETERS OF X264 DEFAULT PRESET

x264core:115r1947baa8ad7					
Preset	Ref	Keyint	Bframes	Scenecut	Lookahead
Medium	3	250	3	40	40

For extensive comparison, well-known x264 and HM are also included. x264 has been widely used in practical applications and is generally treated as one of the best H.264 encoders. It is used in this paper as a baseline for H.264/AVC encoders. We use the default preset (medium) of x264 for our comparison and some key parameters are listed in Table II. For HM, we use the HM 11.0 under the random_access configuration for our comparison, of which the main coding parameters as shown in Table III

TABLE III
KEY PARAMETERS OF HM 11.0 (RANDOM ACCESS)

HM 11.0	
Profile	Main
Level	6.2
MaxCUWidth	64
MaxCUHeight	64
MaxPartitionDepth	4
QuadtreeTULog2MaxSize	5
QuadtreeTULog2MinSize	2
IntraPeriod	32
SearchRange	64
Number of B Slice (GOPSize-1)	3
RDOQ	1
SAO	1
AMP	1

B. 4K Test Sequence

We tested all encoders based on full resolution 4K sequences as they put more press on HEVC encoders and provide more information than low resolution sequences. In fact, we ever found some encoders performed well in normal resolutions (HD and below) but showed abnormal R-D performance with 4K sequences.

It's vitally important to choose proper video sources when conducting encoding comparison. Generally, the performance of algorithms has strong correlations with the video contents. We select test sequences from the database according to texture and motion information. Specifically, we resort to computation-friendly SI (spatial information) and TI (temporal information) to evaluate the content complexity and make the final choice [19]. The SI and TI results of seven selected video sequences are shown in figure 1. To be clear, these test sequences are also listed in Table IV.

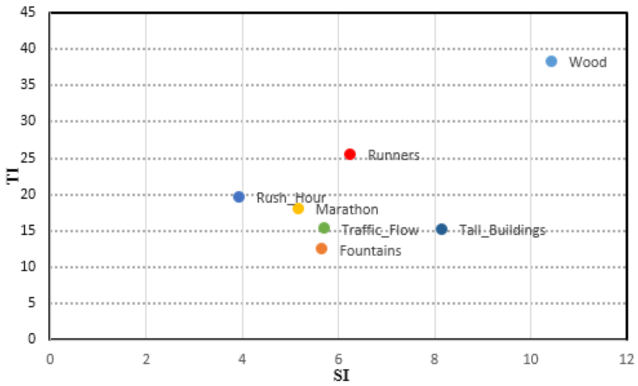


Fig. 1 Spatial Information (SI) and Temporal Information (TI) indexes of the selected video sequences

TABLE IV
TEST SEQUENCES INFORMATION

Sequence	**Resolution**	**Frame rate**
Marathon	3840x2160	30fps
Rush_Hour	3840x2160	30fps
Tall_Buildings	3840x2160	30fps
Traffic_Flow	3840x2160	30fps
Fountains	3840x2160	30fps
Runners	3840x2160	30fps
Wood	3840x2160	30fps

C. Evaluation Criteria

The overall evaluation is based on the PSNR (Peak Signal to Noise Ratio) criterion, which is widely used in video quality assessment and is computed as follows [20].

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \tag{1}$$

$$PSNR = 10*\log_{10}(\frac{(2^n-1)^2}{MSE}) \tag{2}$$

This criterion provides the error between the original video and the reconstructed video. Usually higher PSNR generally indicates higher quality.

In our comparison, different QP values (from 20 to 48) are chosen to achieve different PSNR values.

III. EXPERIMENTAL RESULTS

A. R-D Performances

The following experiments were carried out for each sequence and seven R-D curves are listed below. Four different encoders were chosen and four different presets of x265 were included for the comparison. The overall comparison was conducted on the same platform (Intel Xeon CPU E5-2660 @ 2.20GHz, 16 cores, 24GB physical memory). All videos are raw YUV files in YC_bC_r 4:2:0 format with a length of 10s.

Among the test sequences, the *Fountains* contains many sharp details while *Wood* and *Tall Buildings* have slow motion with an almost static camera and few scene changes.

Other video sequences represent different types of characteristics including high contrast, very fast motion or frequent scene changes.

To span a large bitrate range, we choose QP values from 20 to 48 with QP step equal to 4 (20, 24, 28, 32, 36, 40, 44 and 48). The R-D curves achieved for different video sequences and encoders are presented in Fig.2, where we plot R-D curves based on the luma (Y) component.

In Fig.2, HM and x264 are used respectively as the performance anchors of H.265/HEVC and H.264/AVC for selected videos. As mentioned in the previous section, HM is very slow and generally could be treated as a performance-driven HEVC encoder. In contrast, x264 is highly optimized for both coding performance and speed, and has been widely used for previous study.

In addition, we give widely-used Bjontegaard Distortion-rate (BD-rate) [21] results in Table V by choosing four QP points (28, 32, 36, and 40). From experimental results, we can observe the following facts:

(1) All HEVC encoders have significant gains compared to x264 for 4K video sequences. In specific, DivX265 and *x265_medium* have nearly the same performance as HM in general while x264 has almost 80% bitrate increase compared to HM.

(2) The HEVC encoders perform much better for videos with few scene changes like *Wood* and *Rush_Hour*. On the other hand, gains for sequences with detailed textures like *Fountains* or *Marathon* are limited.

(3) As expected, among four presets of x265, *x265_ultrafast* shows the worst R-D performance (but the fastest coding speed, see the following details). In contrast, *x265_superfast* has approximately 25% bitrate increase compared to *x265_medium. x265_faster* has slightly better R-D performance than *x265_superfast.*

B. Encoding Speed

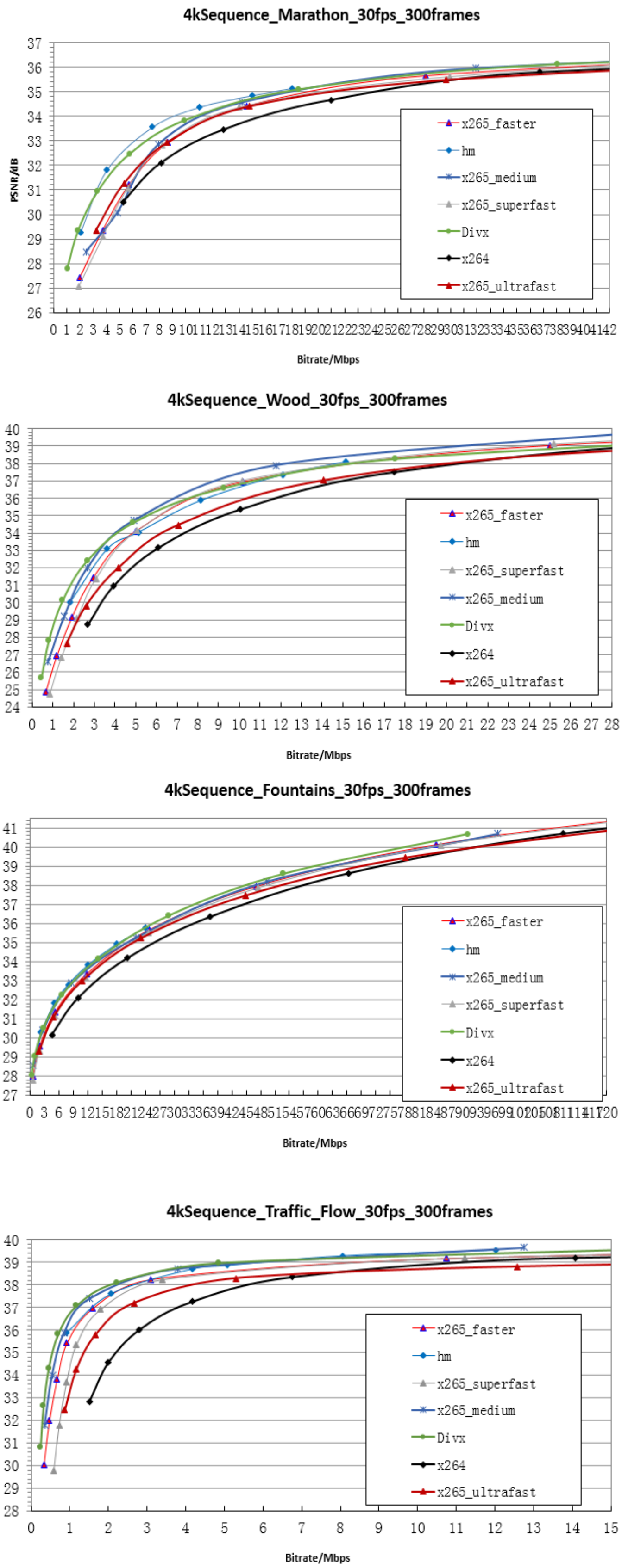
The details of encoding speed can be found in Table VI. From experimental results, we have the following conclusions:

(1) DivX265's coding speed is superior to x265, with a little higher speed than x265_ultrafast for all selected videos. However, around 1 fps speed is still far from that of x264, which is about 12 fps on average.

(2) For each specific encoder, encoding speed is approximately linear with bit rate. High QP (low bit rate) setting is faster than low QP (high bitrate). On the other hand, some sequences are hard for encoders with low coding speed. However, the gap between different sequences is small in general.

(3) For x265 with *superfast* and *faster* presets, the coding speed is between the two extreme presets: *medium* and *ultrafast* as expected.

To highlight the coding speed under different bitrates, the coding speed of the sequence *Runners* is also illustrated in figure 3, where x264 is excluded since it's several times faster than other encoders.



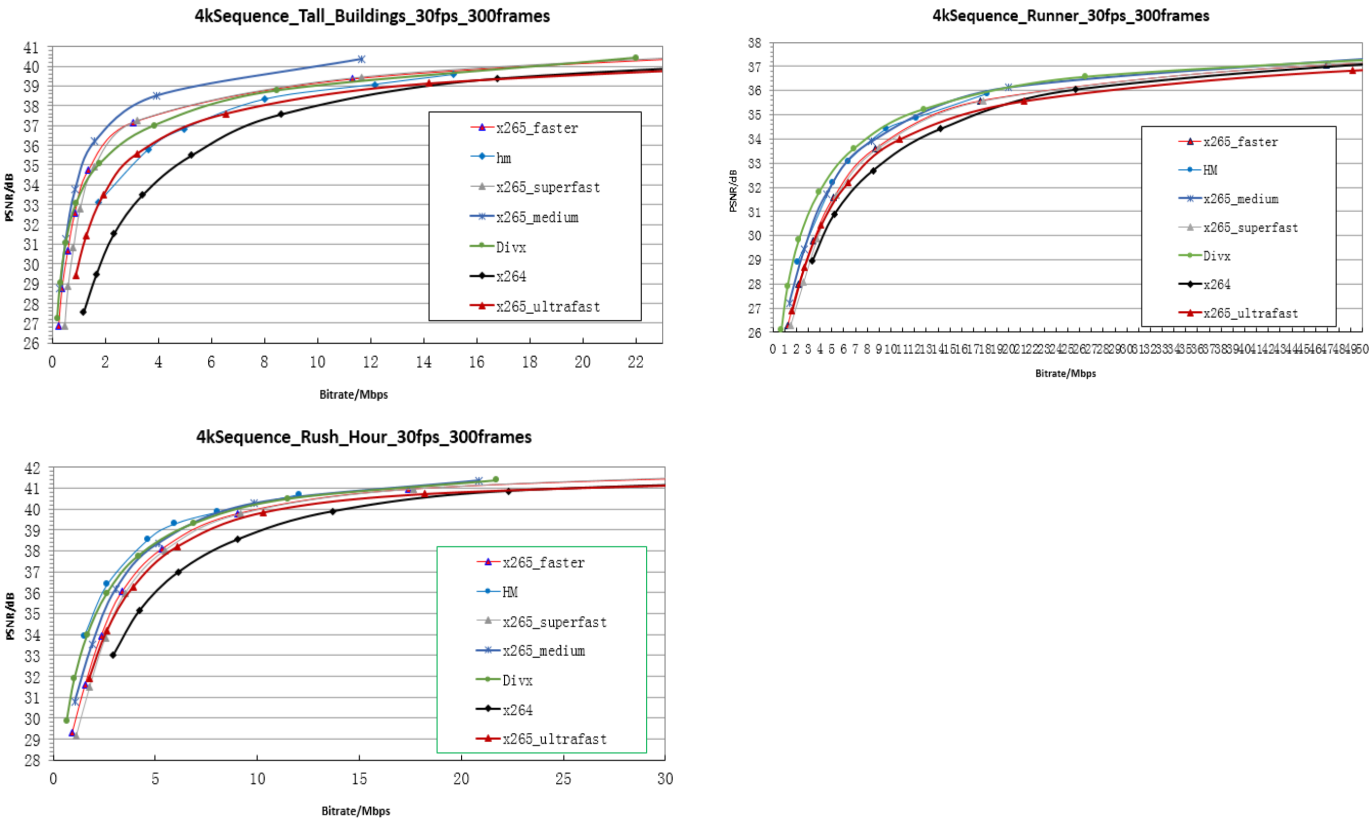


Fig. 2 R-D performance of HM, different presets of x265/x264, and DivX265 on sequences *Rush_Hour*, *Marathon*, *Traffic_Flow*, *Foutains*, *Tall_Buidings, Runners*, and *Wood*

TABLE V
THE BD-BR EXPERIMENTAL RESULTS

Sequences	Bitrate increase(%) under equal PSNR(across the range) based on RD fit					
	x264 vs HM	x265_ultrafast vs HM	x265_superfast vs HM	x265_faster vs HM	x265_medium vs HM	Divx265 vs HM
Runners	48.67	29.07	20.43	17.42	3.18	-7.51
Rush_Hour	90.22	41.01	33.11	26.62	14.06	8.89
Fountains	54.32	23.82	25.36	18.18	6.88	5.58
Marathon	74.91	41.27	39.42	41.82	31.61	15.25
Traffic_flow	162.1	94.33	35.61	22.84	-1.39	-8.86
Tall_Buildings	45.11	-3.38	-45.46	-48.23	-58.91	-31.65
Wood	42.93	25.51	-4.72	-4.36	-17.91	-15.21
Avg	**74.02**	**35.94**	**14.82**	**10.613**	**-3.21**	**-4.787**

TABLE VI
ENCODING SPEED OF DIFFERENT ENCODERS IN TERMS OF FRAMES PER SECOND (fps)

Sequences		x265_medium	x265_faster	x265_superfast	x265_ultrafast	DivX265	x264
Runner	(QP, fps)	(28, 0.08) (32, 0.09) (36, 0.10) (40, 0.11) (44, 0.11) (48, 0.12)	(28, 0.07) (32, 0.17) (36, 0.27) (40, 0.35) (44, 0.42) (48, 0.46)	(28, 0.45) (32, 0.56) (36, 0.65) (40, 0.75) (44, 0.87) (48, 1.01)	(28, 0.54) (32, 0.68) (36, 0.81) (40, 0.93) (44, 1.07) (48, 1.25)	(28, 0.82) (32, 0.88) (36, 1.05) (40, 1.19) (44, 1.35) (48, 1.48)	(28,8.22) (32,10.88) (36,12.05) (40,12.89) (44,12.75) (48,13.26)
	Average fps	0.10	0.29	0.72	0.88	1.13	11.61
Wood	(QP, fps)	(28, 0.08) (32, 0.09) (36, 0.10) (40, 0.11) (44, 0.12) (48, 0.12)	(28, 0.44) (32, 0.56) (36, 0.62) (40, 0.7) (44, 0.85) (48, 0.95)	(28, 0.48) (32, 0.64) (36, 0.74) (40, 0.86) (44, 1.00) (48, 1.23)	(28, 0.63) (32, 0.87) (36, 1.06) (40, 1.23) (44, 1.39) (48, 1.53)	(28, 0.88) (32, 1.00) (36, 1.22) (40, 1.25) (44, 1.37) (48, 1.52)	(28, 9.38) (32,11.27) (36,12.65) (40,12.99) (44,13.17) (48,12.93)
	Average fps	0.10	0.69	0.83	1.12	1.19	12.07
Fountains	(QP, fps)	(28, 0.09) (32, 0.10) (36, 0.12) (40, 0.13) (44, 0.15) (48, 0.16)	(28, 0.23) (32, 0.28) (36, 0.42) (40, 0.61) (44, 0.86) (48, 1.18)	(28, 0.27) (32, 0.37) (36, 0.55) (40, 0.77) (44, 1.07) (48, 1.37)	(28, 0.42) (32, 0.54) (36, 0.76) (40, 1.05) (44, 1.37) (48, 1.76)	(28, 0.83) (32, 0.98) (36, 1.16) (40, 1.42) (44, 1.67) (48, 1.87)	(28, 6.13) (32, 7.89) (36,10.22) (40,13.02) (44,15.18) (48,15.35)
	Average fps	0.13	0.60	0.73	0.98	1.33	11.30
Rush_Hour	(QP, fps)	(28, 0.09) (32, 0.09) (36, 0.11) (40, 0.11) (44, 0.12) (48, 0.14)	(28, 0.16) (32, 0.34) (36, 0.41) (40, 0.47) (44, 0.54) (48, 0.61)	(28, 0.29) (32, 0.40) (36, 0.47) (40, 0.59) (44, 0.71) (48, 0.95)	(28, 0.59) (32, 0.71) (36, 0.83) (40, 0.96) (44, 1.13) (48, 1.32)	(28, 0.95) (32, 1.09) (36, 1.25) (40, 1.41) (44, 1.67) (48, 1.85)	(28, 8.78) (32,10.11) (36,11.03) (40,11.99) (44,12.53) (48,12.79)
	Average fps	0.11	0.42	0.57	0.92	1.37	11.20
Traffic_Flow	(QP, fps)	(28, 0.10) (32, 0.12) (36, 0.14) (40, 0.15) (44, 0.16) (48, 0.17)	(28, 0.59) (32, 0.70) (36, 0.77) (40, 0.83) (44, 0.88) (48, 0.94)	(28, 0.74) (32, 0.90) (36, 1.00) (40, 1.08) (44, 1.13) (48, 1.16)	(28, 0.97) (32, 1.22) (36, 1.37) (40, 1.47) (44, 1.57) (48, 1.64)	(28, 1.02) (32, 1.18) (36, 1.35) (40, 1.59) (44, 1.89) (48, 2.14)	(28,11.54) (32,13.11) (36,14.35) (40,15.01) (44,15.12) (48,15.53)
	Average fps	0.14	0.79	1.00	1.37	1.53	14.11
Tall_Buidings	(QP, fps)	(28, 0.09) (32, 0.11) (36, 0.13) (40, 0.15) (44, 0.16) (48, 0.18)	(28, 0.55) (32, 0.74) (36, 0.92) (40, 0.99) (44, 1.13) (48, 1.18)	(28, 0.97) (32, 1.18) (36, 1.25) (40, 1.31) (44, 1.35) (48, 1.39)	(28,1.05) (32,1.39) (36,1.53) (40,1.61) (44,1.68) (48,1.74)	(28, 0.95) (32, 1.05) (36, 1.16) (40, 1.32) (44, 1.50) (48, 1.69)	(28,10.13) (32,12.34) (36,13.43) (40,14.79) (44,15.41) (48,15.77)
	Average fps	0.14	0.92	1.24	1.5	1.28	13.65
Total Average fps		**0.12**	**0.57**	**0.76**	**1.02**	**1.28**	**11.99**

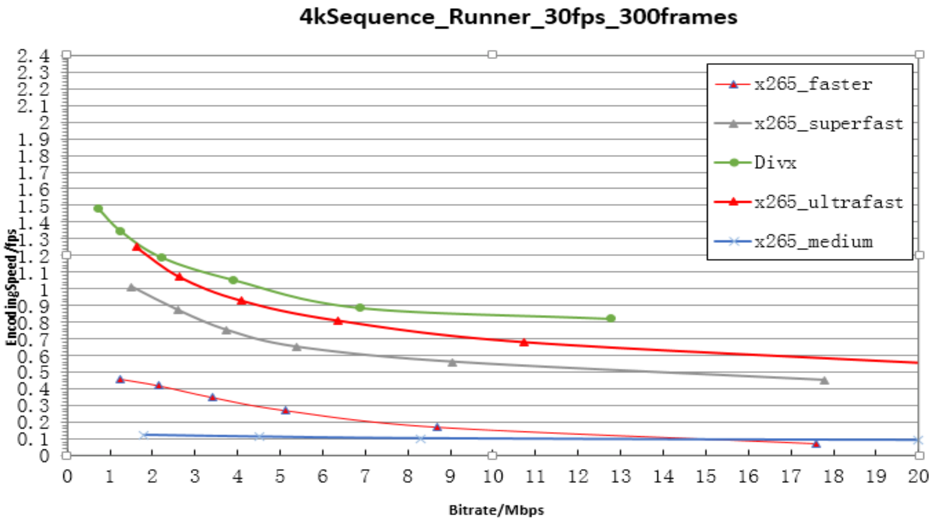


Fig. 3 Encoding speed of different presets of x265 and DivX265 on the *Runners* sequence

IV. Conclusions

The new generation of coding standard− H.265/ MPEG-HEVC will dominate in the coming years. This paper presents a comprehensive comparison of several practical HEVC encoders developed recently in terms of both coding performance and coding speed. Such comparison shows latest HEVC encoders, open source x265 and commercial DivX265, have achieved significant progress compared to the reference codec HM. Specifically, close compression performance has been obtained with an obvious speedup. However, it is also clear that current HEVC encoders are still far from real time 4K encoding and have only one-tenth speed of the optimized H.264/AVC encoder.

With such promising results, we can predict the following trends. First, highly-optimized HEVC encoders will surpass HM in terms of both coding performance and speed soon. Second, real time HEVC encoding for UHD videos (4K or 8K) is still a challenging issue for practical applications in the next few years. There are still lots of optimization works to do to address such challenges. In this paper we only use 4K@30fps sequences; our future work will include higher resolution beyond 4K (8K) and higher frame rate beyond 30 fps (60fps and 120fps, etc.).

Acknowledgment

This work was supported by National 863 project (2012AA011703), National Key Technology R&D Program of China (2013BAH53F04), NSFC (61221001, 61271221), the 111 Project (B07022) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

References

[1] G. J. Sullivan, J.-R. Ohm, “Recent developments in standardization of high efficiency video coding (HEVC)," International Society for Optics and Photonics, pp.77980V-77980V-7, 2010

[2] Haskell, G. Barry, “Digital Video : An Introduction to MPEG-2,” Springer, Sep. 1997.

[3] JVT, “Advanced video coding for generic audiovisual services,” ITU-T Rec. H.264, 2003.

[4] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, “High efficiency video coding (HEVC) text specification draft 9,” document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012.

[5] J. Casajus-Quiros, I. Arambasic,“Video encoder implementation on tilera's tilepro64 multicore processor,” 2013.

[6] Q. Cai, L. Song, G. Li, “Lossy and lossless intra coding performance evaluation: HEVC, H. 264/AVC, JPEG 2000 and JPEG LS,” Proceeding of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012.

[7] G. J. Sullivan, J.-R. Ohm, W. J. Han, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans.Circuits and Systems for Video Technology, pp.1649-1668, 2012.

[8] V. Sze, M. Budagavi, “A comparison of CABAC throughput for HEVC/H.265 VS. AVC/H.264,” Signal Processing Systems (SiPS), pp.165-170, 2013.

[9] M. Zhou, W. Gao, M. Jiang, “HEVC lossless coding and improvements,” IEEE Trans. Circuits and Systems for Video Technology, pp.1839-1843, Dec. 2012.

[10] J.-R. Ohm, G. J. Sullivan, H. Schwarz, et al. “Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC),” IEEE Trans. Circuits and Systems for Video Technology, pp.1669-1684, 2012.

[11] T. Nguyen, D. Marpe, “Performance analysis of HEVC-based intra coding for still image compression,” Picture Coding Symposium (PCS), pp. 233-236, 2012.

[12] D. Grois, D. Marpe, A. Mulayoff, “Performance comparison of H. 265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders,” Picture Coding Symposium (PCS), pp.394-397, 2013.

[13] J. Vanne, M. Viitanen, T.D. Hamalainen, “Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs,” IEEE Trans. Circuits and Systems for Video Technology, pp.1885-1898, 2012.

[14] x265 Latest Reference Software [Online]. Website: http://x265.org/

[15] DivX265 Latest Reference Software [Online]. Website: http://labs. divx.com/Divx265
[16] SJTU 4K Video Sequences [Online]. Website: http://medialab.sjtu.edu.cn/web4k/index.html
[17] L. Song, X. Tang, W. Zhang, X. Yang, P. Xia, "The SJTU 4K Video Sequence Dataset", The Fifth International Workshop on Quality of Multimedia Experience (QoMEX 2013) , Klagenfurt am Worthersee, Austria, Jul., 2013.
[18] x264 Latest Reference Software [Online] Website: http://www.videolan.org/developers/x264.html
[19] M. N. Garcia, A. Raake, “Towards content-related features for parametric video quality prediction of IPTV services,” ICASSP, pp.757-760, 2008.
[20] B. Li, G. J. Sullivan, and J. Xu, “Comparison of compression performance of HEVC working draft 4 with AVC high profile,” Doc. JCTVC-G399, Nov. 2011.
[21] G. Bjontegaard, “Improvements of the BD-PSNR model,” VCEG-AI11, July 2008.