Performance Evaluation of Kvazaar HEVC Intra Encoder on Xeon Phi Many-core Processor

Ari Koivula, Marko Viitanen, Ari Lemmetti, Jarno Vanne, Timo D. Hämäläinen
Department of Pervasive Computing
Tampere University of Technology
Tampere, Finland

Abstract—**This paper analyzes parallel scalability and coding speed of our open-source Kvazaar HEVC intra encoder on Intel Xeon Phi 61-core coprocessor that supports up to four hardware threads per core. The evaluated parallelization schemes of Kvazaar are 1) Wavefront Parallel Processing (WPP); and 2) tiles, both accelerated with picture-level parallel processing. With WPP, the C implementation of Kvazaar high-quality preset achieves an average speedup of 1.3 and a bit rate gain of 0.7% over the respective implementation of x265. Using tiles makes Kvazaar 1.4 times faster than x265 but at a cost of 0.3% bit rate loss. When high-speed presets are used, the speedup of Kvazaar increases to 1.4 with WPP and to 1.9 with tiles. Moreover, the respective coding efficiency of Kvazaar rises to 11.2% and 10.3%. Kvazaar also scales almost linearly to the number of cores in the processor. Even if the peak coding speed of Kvazaar on Xeon Phi is lower than that on the Intel 8-core i7 processor, our parallel scalability results promise excellent speed for Kvazaar on massively parallel processors equipped with more powerful cores.**

Keywords—HEVC; Kvazaar HEVC intra encoder; Wavefront Parallel Processing (WPP); tiles; Xeon Phi many-core processor

I. Introduction

High Efficiency Video Coding (*HEVC*) standard [1], [2] represents the state-of-the-art in video coding. It has been developed by *Joint Collaborative Team on Video Coding* (*JCT-VC*) as a joint activity of *ITU-T Video Coding Experts Group* (*VCEG*) and *ISO/IEC Moving Picture Experts Group* (*MPEG*). HEVC is published as twin text by ITU, ISO, and IEC as ITU-T H.265 | ISO/IEC 23008-2. Its first edition contains three profiles: *Main*, *Main Still Picture*, and *Main 10* profiles.

This paper focuses on HEVC Main Profile under *all-intra* (*AI*) coding [3] configuration. HEVC intra coding is shown to improve coding efficiency by 23% over the current mainstream standard AVC [4] for the same objective quality but at a cost of over 3× encoding complexity [5]. HEVC compensates this complexity overhead by specifying two principal, mutually exclusive, data-level parallelization strategies [6] for intra coding: 1) *Wavefront Parallel Processing* (*WPP*) [7] and 2) *tiles* [8]. They are by far the most popular schemes for implementing parallel processing in software HEVC encoders.

Currently, there exist three noteworthy open-source HEVC encoders that all support parallel processing: x265 [9], f265 [10], and our Kvazaar [11]. x265 offers WPP but not tiles whereas f265 supports tiles but not WPP. Kvazaar offers both tiles and WPP [12]. Kvazaar and x265 also accelerate parallel coding further with *picture-level parallel processing*. This paper focuses mainly on Kvazaar and benchmarks it over x265. f265 is excluded from our evaluations since the project seems not to be under active development anymore. Parallelization of commercial software HEVC encoders is not considered here either due to their confidential implementation details.

WPP and tiles are realized in practice by multithreading the software encoder on a multi or many-core processor where data processing flows are identical to each core. Our previous work [12] introduces parallelization strategies of Kvazaar for multi-core processors of up to eight cores. This paper extends Kvazaar parallelization to many-core processors with tens or even hundreds of cores. For this purpose, the most potential many-core architectures include Intel Xeon Phi processor family [13], Kalray's *Multi-Purpose Processor Architecture* (*MPPA*) [14], and Nvidia Tesla *Graphics Processing Unit* (*GPU*) [15] out of which Xeon Phi was selected for this study.

The rest of this paper is organized as follows. Section 2 presents the parallelization schemes of Kvazaar intra encoder. The essential features of Intel Xeon Phi 7120X coprocessor are detailed in Section 3. Section 4 benchmarks parallel scalability of Kvazaar intra coding on Xeon Phi and compares the results with those of x265 as a function of the encoding threads (from 16 to 240). Section 5 gives the conclusion.

II. Parallelization of Kvazaar Intra Encoder

Kvazaar intra encoder supports HEVC Main profile for 8-bit 4:2:0 video with two presets: 1) RD1 for high-speed encoding; and 2) RD2 for high-quality encoding. A more detailed description of these presets and the overall intra coding scheme of Kvazaar is given in [16].

For parallel encoding, Kvazaar offers three schemes: tiles, WPP, and picture-level parallel processing [12]. WPP and tiles divide a picture into multiple partitions that can be processed in parallel. The partitions are composed of an integer number of *coding tree units* (*CTUs*) [2] of size 64 × 64, 32 × 32, or 16 × 16 pixels depending on Kvazaar parametrization. Picture-level parallel processing can be utilized jointly with tiles and WPP.

Kvazaar parallelization is implemented using a thread pool with a single CTU as the smallest work unit. The CTUs are put in a queue in the order in which they would be processed in a single threaded case, and the free worker threads always select the first CTU with no dependencies for processing.

A. *Wavefront Parallel Processing (WPP)*

In WPP [7], the partition size equals a single CTU row, so the maximum number of available partitions equals the number of CTU rows. However, wavefront dependencies prevent all partitions from being started simultaneously. The first CTU row of the picture is started immediately, but the latter rows cannot be started until two CTUs have been processed in the former row. The same start condition holds for all CTUs in a row, i.e., the processing of the former row has to be two CTUs ahead of the latter through the whole row. Since the processing time varies between CTUs, the start condition makes the optimal processing order of CTUs less predictable. Kvazaar compensates this by tracking spatial dependencies of CTUs and assigning them to worker threads accordingly. A worker thread can select any CTU that has no unsatisfied dependencies.

B. *Tiles*

Tiles [8] segment the picture into the rectangular groups of CTUs. The number of tiles and the location of their boundaries can be specified either at picture or sequence level. In this paper, the selected tile configuration is 5 × 3, i.e., each picture is divided into 5 tiles horizontally and 3 vertically. Kvazaar processes tiles independently, so the parallelization overhead is diminutive, but the bit rate grows together with the tile count.

C. *Picture-level Parallel Processing*

In Kvazaar, picture-level parallelization is integrated with WPP and tiles. Because there are no dependencies between pictures in intra coding, the scheduler is free to select jobs from the next picture if necessary.

Kvazaar selects the number of parallel processed pictures based on the dimensions of the pictures and the thread count. The larger and wider the picture, the more WPP threads can be processed in parallel. The number of parallel pictures is derived from the largest number of threads per picture so that all threads can work simultaneously for at least 85% of the picture. This constraint reduces the number of parallel pictures without sacrificing encoding speed. For tiles, the number of pictures is selected so that there are four times as many tiles as threads.

III. Intel Xeon Phi

The evaluated Intel Xeon Phi coprocessor is based on Intel *Many Integrated Core* (*MIC*) architecture [17] in which multiple x86-based processor cores are placed in a single chip for massively parallel computing. Xeon Phi is connected to a host processor via PCI Express bus. It supports either an offload programming model as general-purpose GPUs or a program code can be recompiled and run natively on it.

Table I tabulates the specifications of the Xeon Phi 7120X coprocessor benchmarked in this paper. The chip belongs to the first-generation Xeon Phi family, codenamed Knights Corner, announced in 2011 on 22 nm process. It is equipped with 61 cores each of which contains in-order execution pipeline and hardware support for 4-way simultaneous multithreading. Each core incorporates a vector processing unit with 512-bit wide registers and a unique SIMD instruction set extension called *Initial Many Core Instructions* (*IMCI*) for vector calculations. Other SIMD extensions like SSE or AVX are not supported, but the Intel compiler is dedicated to utilize IMCI efficiently.

TABLE I. Intel Xeon Phi platform [13]

Processor	Intel Xeon Phi 7120X (61 × 1.238 GHz)
Memory	16 GB
L1 cache	61 × 32 KB (instruction) + 61 × 32 KB (data)
L2 cache	61 × 512 KB
Compiler	Intel C++ Compiler 2013.3.163
Operating system	64-bit Linux 2.6 µOS with busybox

The chip contains a 16 GB GDDR5 memory. Each core has a separate 32 KB L1 instruction and data caches and a dedicated 512 KB L2 cache. Memory accesses need to be aligned to 64 byte cache lines for the maximum performance. Interfaces to PCIe bus, memory, and the caches are all connected by a bidirectional ring interconnect.

IV. Evaluation of Kvazaar and x265 on Xeon Phi

The complexity characteristics of Kvazaar version v0.5.0 [11] were evaluated on Xeon Phi (Table I) by benchmarking its RD1 (Kvazaar$_{rd1}$) and RD2 (Kvazaar$_{rd2}$) presets against x265 version 1.6 [9]. The evaluated presets of x265 are veryslow (x265$_{veryslow}$) and ultrafast (x265$_{ultrafast}$). They represent the slowest and fastest practical presets of x265, respectively.

Table II lists the command line parameters used in our encoder tests. Release versions of Kvazaar and x265 were compiled with Intel C and C++ Compilers with –mmic, –O3, -DNDEBUG, and -unroll0 options using the build files of the projects. For Kvazaar, -fno-inline-functions was additionally enabled to fix compiler bugs caused by –O3, but this option was left out for x265 due to its negative impact on x265 speed. No other Xeon Phi specific optimizations were made.

A. *Analysis Setup*

Table III tabulates the 8-bit test sequences applied in our experiments. The 2160p sequences are available online on our website [18] whereas the 1600p and 1080p sequences are from HEVC common test conditions (classes *A*-*B*) [19].

The complexity comparison between Kvazaar and x265 is based on their encoding times. The average coding speeds were measured separately for each format and for the *quantization parameter* (*QP*) values of 22 and 37. For a reliable comparison, only one encoder instance was run at a time. All input files were copied to Xeon Phi and the binaries were executed natively by connecting to the card with SSH.

The bitrate differences between the encoder presets have been compared in terms of the *Bjøntegaard delta bitrate* (*BD-rate*) [20]. The RD curves for the BD-rate computations have been interpolated with the piecewise cubic interpolation [21] through experimentally specified RD points that represent the QPs of 22, 27, 32, and 37. In the evaluated cases, the BD-rate is independent on the number of threads, so the average BD-rate per sequence is computed only once for each encoder instance.

B. *Parallel Scalability Analysis*

Fig. 1 plots the coding speeds of Kvazaar and x265 as a function of threads on Xeon Phi that contains 61 physical and 244 logical cores. The parallel scalability has been analyzed with 16 to 240 threads in 16 thread steps to find the peak speed. The average speed curves with the QP value of 22 are shown for the 2160p, 1600p, and 1080p formats in Fig 1 (a), (c), and

TABLE II. ENCODER COMMAND LINE PARAMETERS

Encoder	Parameters
Kvazaar	--input-res=(res) -n (frames) --cpuid=0 -p 1 -q (qp) --owf=31 --rd=(preset) --threads=(threads) --no-transform-skip
wpp	--wpp
tiles	--tiles-width-split=u(x tiles) --tiles-height-split=u(y tiles)
x265	--tune psnr --psnr --hash 3 --log-level debug --no-progress --frame-threads 16 -I 1 --no-open-gop --no-scenecut --rc-lookahead 0 --lookahead-slices 0 --bframes 0 --fps 30 --pmode --pool (threads) --preset (preset) --frames (frames) -q (qp) --ipratio 1 --input-res (res)

(e), respectively. The remaining charts plot the corresponding results for the QP value of 37. For readability, the curves are linked to the associated encoder presets with symbols (*A-F*).

Kvazaar$_{\text{rd1}}$ with tiles achieves the highest coding speed in all evaluated cases. However, the benefit of additional threads is under 5% after 160 threads at QP = 22 and 128 threads at QP = 37. In some cases, the speed does not only saturate to the maximum, but starts decreasing. Similar findings are reported in [22] where affinity problems are mentioned as a possible reason for the performance drop. Another suggested reason for the speed degradation is the limited bandwidth between memory and 61 processor cores on Xeon Phi.

Kvazaar$_{\text{rd1}}$ with WPP is the second fastest preset and its peak performance is reached at 128 threads at the latest. At QP = 37, the maximum is met already at 64 threads with 2160p format. Even though x265$_{\text{ultrafast}}$ is faster than Kvazaar$_{\text{rd1}}$ with WPP in occasional cases, it encounters serious fluctuation in performance. Temporary slowdown occurs, e.g., with thread counts of 80 and 144. In these cases, a thread count exceeds the limit of a thread pool size specified in the source code of x265.

In all test cases, Kvazaar$_{\text{rd2}}$ is faster than x265$_{\text{veryslow}}$ that encounters similar speed degradations as x265$_{\text{ultrafast}}$ with thread counts of 80, 144, and 208. Kvazaar$_{\text{rd2}}$ achieves the peak performance with tiles but its coding speed remains almost the same with WPP. With tiles and WPP, the speed of Kvazaar$_{\text{rd2}}$ improves less than 5% after 176 threads.

In summary, all presets of Kvazaar scale almost linearly to the number of physical cores in the processor. Kvazaar can also take advantage of logical cores, but the speedup decreases gradually as a function of the logical core count.

C. Rate-distortion-complexity Analysis

Table IV reports the average format-specific coding speeds and BD-rates of Kvazaar$_{\text{rd2}}$ over that of x265$_{\text{veryslow}}$. For each format, the reported thread count is used for Kvazaar$_{\text{rd2}}$ since the maximum average coding speed for the QP values of 22 and 37 is obtained with it. The corresponding thread counts for x265$_{\text{veryslow}}$ are 192, 176, and 128.

With WPP, Kvazaar$_{\text{rd2}}$ obtains an average speedup of 1.3× (from 1.2× to 1.4×) over x265$_{\text{veryslow}}$ and improves BD-rate by 0.7% (from -1.8% to 0.1%). Switching to tiles speeds up Kvazaar$_{\text{rd2}}$ a bit more but at cost of an average BD-rate penalty of 0.3% (from -0.6% to 0.5%) over x265. Therefore, Kvazaar$_{\text{rd2}}$ with WPP is the most recommended high-quality setting.

Table V tabulates the respective speedups and BD-rates for the high-speed presets. For x265$_{\text{ultrafast}}$, the best coding speed is obtained using 64 threads in all formats. In this case, the average coding speed of Kvazaar$_{\text{rd1}}$ with WPP is 1.4× (from 1.3× to 1.5×) higher than that of x265$_{\text{ultrafast}}$ and its BD-rate gain is as much as 11.2% (from -8.7% to -13.5%). Using tiles raises the speedup of Kvazaar$_{\text{rd1}}$ to 1.9× (from 1.8× to 2.1×) while its BD-rate is still 10.3% (from -7.9% to -12.8%) better.

TABLE III. 8-BIT TEST SEQUENCES

Format	Sequence	# of frames	Frame rate
3840×2160 (2160p)	Beauty	600	120 fps
	HoneyBee	600	120 fps
	Jockey	600	120 fps
2560×1600 (1600p)	Traffic	150	30 fps
	PeopleOnStreet	150	30 fps
1920×1080 (1080p)	Kimono	240	24 fps
	ParkScene	240	24 fps
	BasketballDrive	500	50 fps

Our results show that replacing WPP with tiles has quite similar impact on BD-rate in the high-quality and high-speed cases. However, Kvazaar$_{\text{rd1}}$ with tiles is the recommended setting in this latter case due to its positive impact on speed.

D. Feasibility Analysis

Although Kvazaar is clearly faster than x265 in the evaluated cases, its performance is far from real-time coding. Table VI tabulates the coding speeds of the selected settings of Kvazaar on Xeon Phi and on the 3.0 GHz Intel 8-core i7 processor. For a fair comparison, the same test set (Table III) is also executed with i7 using 16 threads with only compiler SIMD optimizations enabled.

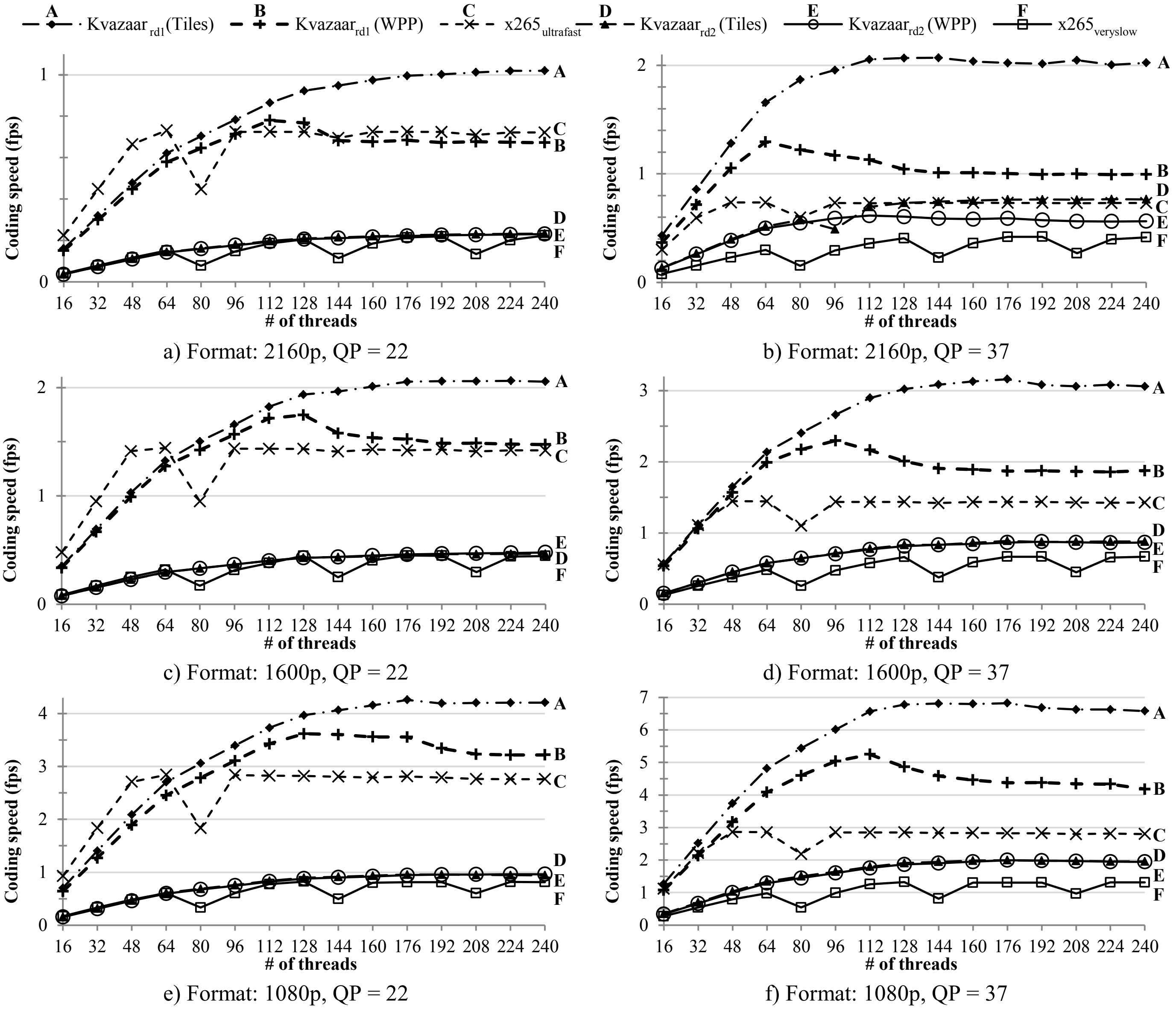
On Xeon Phi, Kvazaar$_{\text{rd2}}$ with WPP is able to encode 2160p, 1600p, and 1080p sequences at 0.4, 0.7, and 1.5 fps, respectively. On i7, the same test set is encoded 13-30% faster. The analogous results are reported for Kvazaar$_{\text{rd1}}$ with tiles whose execution on i7 is 19-35% faster than on Xeon Phi.

Our results show that Xeon Phi is not an optimal platform for Kvazaar. However, the results validate that Kvazaar scales almost linearly to the number of physical cores in the processor and can also get additional gain from hardware threads. Hence, real-time encoding can be anticipated on many-core processors equipped with more efficient and better utilizable processor cores.

V. CONCLUSIONS

This paper analyzed feasibility and parallel scalability of our Kvazaar HEVC intra encoder on Xeon Phi 7120X 61-core coprocessor. The examined HEVC parallelization strategies were WPP and tiles, both accelerated with picture-level parallel processing. Our rate-distortion-complexity results recommend using WPP with high-quality preset and tiles with high-speed preset of Kvazaar. They were measured to attain 0.7% and 10.3% bit rate gains over x265, respectively.

Even though Kvazaar is also shown to be 1.3-1.9× faster than x265, its overall performance on Xeon Phi is still far from real-time 1080p encoding. However, the obtained parallel scalability results are promising, so the real-time speed is on the horizon if Kvazaar is mapped to a more powerful many-core processor.



a) Format: 2160p, QP = 22

b) Format: 2160p, QP = 37

c) Format: 1600p, QP = 22

d) Format: 1600p, QP = 37

e) Format: 1080p, QP = 22

f) Format: 1080p, QP = 37

Fig. 1. Coding speeds of Kvazaar and x265 on Xeon Phi many-core processor as a function of the encoding threads.

TABLE IV. KVAZAAR VS. X265: HIGH-QUALITY PRESETS

Format	Kvazaar$_{rd2}$ (WPP)			Kvazaar$_{rd2}$ (Tiles)		
	Threads	Speedup	BD-rate	Threads	Speedup	BD-rate
2160p	128	1.3×	-1.8%	240	1.6×	-0.6%
1600p	192	1.2×	0.1%	176	1.2×	0.9%
1080p	176	1.4×	-0.5%	176	1.4×	0.5%
Average	-	1.3×	-0.7%	-	1.4×	0.3%

TABLE V. KVAZAAR VS. X265: HIGH-SPEED PRESETS

Format	Kvazaar$_{rd1}$ (WPP)			Kvazaar$_{rd1}$ (Tiles)		
	Threads	Speedup	BD-rate	Threads	Speedup	BD-rate
2160p	112	1.3×	-8.7%	208	2.1×	-7.9%
1600p	112	1.3×	-13.5%	176	1.8×	-12.8%
1080p	112	1.5×	-11.4%	176	1.9×	-10.2%
Average	-	1.4×	-11.2%	-	1.9×	-10.3%

TABLE VI. CODING SPEED OF KVAZAAR ON XEON PHI AND I7

Format	Kvazaar$_{rd2}$ (WPP)		Kvazaar$_{rd1}$ (Tiles)	
	Xeon Phi	Core i7	Xeon Phi	Core i7
2160p	0.4 fps	0.5 fps	1.5 fps	2.1 fps
1600p	0.7 fps	0.8 fps	2.6 fps	3.1 fps
1080p	1.5 fps	1.7 fps	5.5 fps	6.6 fps

ACKNOWLEDGEMENT

This work was supported in part by the European Celtic-Plus Project 4KREPROSYS. The authors would also like to thank all contributors of our Kvazaar open-source project [11].

REFERENCES

[1] *High Efficiency Video Coding*, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, Apr. 2013.

[2] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1649-1668.

[3] J. Lainema, F. Bossen, W. J. Han, J. Min, and K. Ugur, “Intra coding of the HEVC standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1792-1801.

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

[5] J. Vanne, M. Viitanen, T. D. Hämäläinen, and A. Hallapuro, “Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1885-1898.

[6] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, “Parallel scalability and efficiency of HEVC parallelization approaches,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, Dec. 2012, pp. 1827-1838.

[7] F. Henry and S. Pateux, “Wavefront parallel processing,” *Document JCTVC-E196*, Geneva, Switzerland, Mar. 2011.

[8] K. Misra, A. Segall, M. Horowitz, X. Shilin, A. Fuldseth, and Z. Minhua, “An overview of tiles in HEVC,” *IEEE J. Select. Topics in Signal Process.*, vol. 7, no. 6, Dec. 2013, pp. 969-977.

[9] *x265* [Online]. Available: http://x265.org/

[10] f265 [Online]. Available: http://f265.org/

[11] *Kvazaar HEVC encoder* [Online]. Available: https://github.com/ultravideo/kvazaar

[12] A. Koivula, M. Viitanen, J. Vanne, T. D. Hämäläinen, and L. Fasnacht, “Parallelization of Kvazaar HEVC intra encoder for multi-core processors,” *in Proc. IEEE Workshop Signal Process. Syst.*, Hangzhou, China, Oct. 2015.

[13] *Intel Xeon Phi Product Family* [Online]. Available: www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.html?wapkw=xeon+phi

[14] B. D. de Dinechin, R. Ayrignac, P. E. Beaucamps, P. Couvert, B. Ganne, P. G. de Massas, F. Jacquet, S. Jones, N. M. Chaisemartin, F. Riss, and T. Strudel, “A clustered manycore processor architecture for embedded and accelerated applications,” *IEEE High Performance Extreme Computing Conference*, Sep. 2013, pp.1-6.

[15] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, “NVIDIA Tesla: a unified graphics and computing architecture,” *IEEE Micro*, vol.28, no.2, Mar/Apr 2008, pp. 39-55.

[16] M. Viitanen, A. Koivula, A. Lemmetti, J. Vanne, and T. D. Hämäläinen, “Kvazaar HEVC encoder for efficient intra coding,” *in Proc. IEEE Int. Symp. Circuits Syst.*, Lisbon, Portugal, May 2015, pp. 1662-1665.

[17] *Intel® Xeon Phi™ Coprocessor System Software Developers Guide*, Intel, Mar 2014.

[18] *Ultra video group* [Online]. Available: http://ultravideo.cs.tut.fi/

[19] F. Bossen, “Common test conditions and software reference configurations,” *Document JCTVC-J1100*, Stockholm, Sweden, Jul. 2012.

[20] G. Bjøntegaard, “Calculation of average PSNR differences between RD curves,” *Document VCEG-M33*, Austin, TX, USA, Apr. 2001, pp. 1-4.

[21] J. Wang, X. Yu, and D. He, “On BD-rate calculation,” *Document JCTVC-F270*, Torino, Italy, Jul. 2011.

[22] J. V. F Lima, F. Broquedis, T. Gautier, and B. Raffin, “Preliminary experiments with XKaapi on Intel Xeon Phi coprocessor,” *in Proc. Int. Symp. Computer Architecture and High Performance Computing*, Oct. 2013, pp. 105-112.