Advances in Engineering Software 000 (2016) 1–13

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Synchronous and asynchronous HEVC parallel encoder versions based on a GOP approach

H. Migallón[a,*], J.L. Hernández-Losada[b], G. Cebrián-Márquez[b], P. Piñol[a], J.L. Martínez[b], O. López-Granado[a], M.P. Malumbres[a]

[a] Department of Physics and Computer Architecture, University Miguel Hernández, Elche, E-03202 Alicante, Spain
[b] Albacete Research Institute of Informatics, Universidad de Castilla-La Mancha, E-02071 Albacete, Spain

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Parallel algorithms
Video coding
HEVC
Multicore
Performance
GOP-based algorithms
Synchronous algorithms
Asynchronous algorithms

ABSTRACT

In this paper, we focus on applying parallel processing techniques to HEVC encoder in order to significantly reduce the computational power requirements without disturbing its coding efficiency. So, we propose several, synchronous and asynchronous, parallelization approaches working at a coarse grain parallelization level, based on the Group Of Pictures (GOP), which we call GOP-based level. GOP-based approaches encode simultaneously several groups of consecutive frames. Depending on how these GOPs are conformed and distributed it is critical to obtain good parallel performance . The results show that near ideal efficiencies are obtained using up to 10 cores. Furthermore, when the computational load is unbalanced, the asynchronous versions outperform the synchronous ones. The parallel algorithms developed in this work support all standard coding modes proposed by the reference software.

1. Introduction

Recently, the Joint Collaborative Team on Video Coding (JCT-VC) co-established by ISO/IEC MPEG (Motion Experts Group) and ITU-T VCEG (Video Coding Experts Group), has standardized the next-generation video coding technology called High Efficiency Video Coding (HEVC) [1]. This new standard will replace the current H.264/AVC (Advanced Video Coding) [2] standard in order to deal with nowadays and future multimedia market trends, since 4K definition video content is a nowadays fact and 8K definition video will not take too long to become a reality. Even more, the new standard supports high quality color depth at 8 and 10 bits. HEVC greatly improved the coding efficiency over its predecessor (H.264/AVC) by a factor of almost twice while maintaining an equivalent visual quality [3].

Regarding complexity, in [4], Bossen et al. studied the complexity aspects of HEVC encoding and decoding software. This study concludes that the encoding process is much more challenging than the decoding process, e.g., encoding one second of a 1080p60 HD (High Definition) video with the reference software encoder can take longer than one hour when running in an off-the-shelf desktop computer. Therefore, HEVC encoder optimization will be a hot research topic in years to come.

Several works about complexity analysis and parallelization strategies for the emerging HEVC standard can be found in the literature [4–6]. Most of parallelization proposals are focused in the decoding side, looking for the most appropriate parallel optimizations at the decoder that provide real-time decoding of High-Definition (HD) and Ultra-High-Definition (UHD) video contents. In [7,8] the authors present a variation of Wavefront Parallel Processing (WPP) called Overlapped Wavefront (OWF) for the HEVC decoder in which the executions over consecutive pictures are overlapped. In a multi-threaded approach of the HEVC decoder, a picture is decoded by several threads at the same time, being each thread in charge of decoding different Coding Tree Block (CTB) rows. In these works, authors claim that a single thread may continue processing the next picture when it finishes the current one, without waiting for the other threads. These variations allow a better parallel processing efficiency, reducing the overall decoding time. Recently, in [9] the authors mixed tiles, WPP and SIMD (Single-Instruction Multiple-Data instruction set extension to the x86 architecture) instructions to develop a real-time HEVC decoder.

At the moment, only a few works focused on the HEVC encoder have been reported. In [10] authors propose a fine-grain parallel optimization of the HEVC motion estimation module that performs at the same time the motion prediction of all Prediction Units (PUs) available at one Coding Unit (CU). In [11,12] authors propose a real-time motion estimation block over focusing on the optimization of motion estimation algorithms using an FPGA-based low cost embedded system with a combination of synchronous

* Corresponding author. Tel.: +34 966658390; fax: +34 966658814.
E-mail address: hmigallon@umh.es (H. Migallón).

http://dx.doi.org/10.1016/j.advengsoft.2016.01.020
0965-9978/

dynamic random access memory (SDRAM) with on-chip memory of software-based Nios II processors. Through the optimizations of memory access in this platform, time savings of up 53% were achieved in the motion estimation module. In [13] authors propose a parallelization at the intra prediction module that consists on removing data dependencies among subblocks of a CU, obtaining interesting speed-up results with a negligible loss in coding performance. Other recent works are focused on changes in the scanning order. For example, in [14] the authors propose a CU scanning order based on a diamond search obtaining a good scheme for massive parallel processing. Also in [15] the authors propose to change the HEVC deblocking filter processing order obtaining time savings of 37.93% over many-core architectures.

In this paper, we will focus on applying parallel processing techniques to the HEVC encoder in order to significantly reduce the computational power requirements without disturbing the coding efficiency. Instead of focusing the optimization on one specific module of the HEVC encoder, as other proposals do, our proposals use OpenMP programming paradigm working at a coarse grain parallelization level which we call GOP-based level. GOP-based approaches encode simultaneously several Group Of Pictures (GOP). Depending on how these GOPs are conformed and distributed it is critical to obtain good parallel performance, taking also into account the level of coding efficiency degradation. This paper is based upon Migallón et al. [16], including more results and additional research such (a) new asynchronous parallel versions, and (b) a comparison between synchronous and asynchronous parallel versions in terms of computational complexity (coding time) and speed-up.

The remainder of this paper is organized as follows, in Section 2 an overview of the available profiles and parallel strategies in HEVC are presented. Sections 3 and 4 describe the GOP-based parallel alternatives proposed for both synchronous and asynchronous architectures , while in Section 5 a comparison between the proposed parallel approaches is presented. Finally, in Section 6 some conclusions are drawn.

2. HEVC coding modes and parallel strategies

HEVC follows a hybrid video coding scheme consisting on a sequence of three main steps. First, the spatial or temporal redundancy is exploited to make a prediction of a frame region and, in this way, only the residuum of the prediction and some side information will be encoded. In the second step, the residuum is transformed into the frequency domain and the resulting coefficients are quantized (lossy compression). Depending on the quantization step, we will achieve a higher or lower compression ratio in the bit stream, and the reconstructed video sequence will exhibit a higher or lower visual quality. In the third step, entropy coding is applied to the quantized coefficients and the side information in order to further compress the bit stream.

In the encoding process, each frame is divided into small square regions called Coding Units (CU). Spatial redundancy is exploited by obtaining a prediction for a CU (or CU subpartitions) using the nearby pixels in the same frame. Temporal redundancy is exploited by searching for similar CUs (or CU subpartitions) in previous or past frames in order to obtain a prediction.

In our tests we have used the available coding modes of the reference software package [17]: All Intra (AI), Random Access (RA), Low-Delay B (LB), and Low-Delay P (LP). Each mode has different characteristics and can be used in different situations, depending on the requirements of each application.

In All Intra mode every frame is coded as an I-frame, i.e., it is encoded without any motion estimation/compensation. So, each frame is independent from the other frames in the sequence. This mode gets lower compression rates compared to the other 3 modes, but the encoding process is faster. Fig. 1 shows the AI mode coding structure.

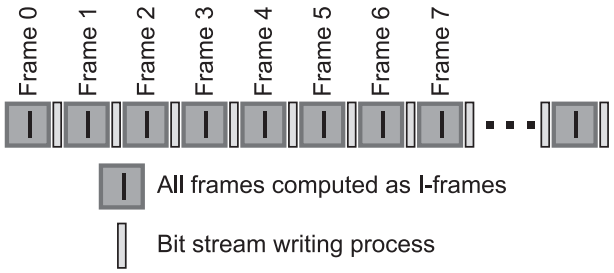

Fig. 1. All Intra (AI) mode sequential structure.

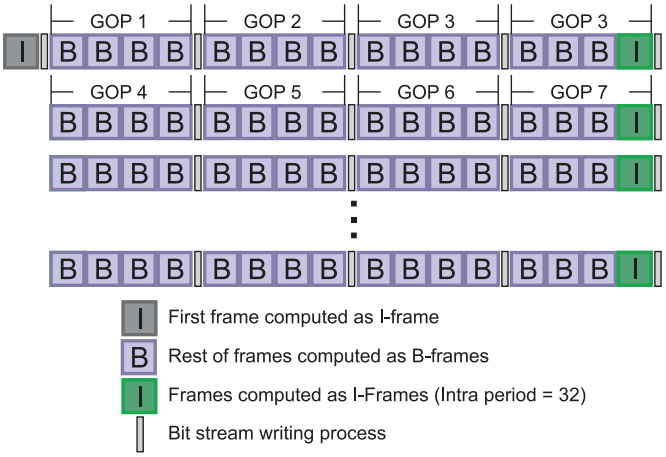

Fig. 2. Random Access (RA) mode sequential structure.

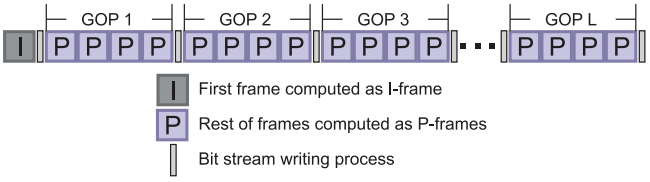

Fig. 3. Low delay P (LP) mode sequential structure.

Random Access mode combines I-frames and B-frames. In this mode, reference frames used to perform the motion estimation process can be located earlier or later than the frame we are currently encoding. So, encoding (and decoding) order is not the same as rendering order. In the HEVC standard, the RA mode uses a GOP size of 8 frames. To allow navigating through the coded sequence (pointing to a certain video sequence frame) or to allow trick modes like fast forward, an I-frame is inserted periodically. The intra refresh period must be a multiple of GOP size. Fig. 2 shows the coding structure for the RA mode when the intra refresh period is equal to 32 frames. The bit stream updating is performed after each GOP computation.

Low-Delay modes (LP and LB) encode each frame in rendering order. First an I-frame is inserted in the coded bit stream and then only P-frames (or B-frames) are used for the rest of the sequence, with a GOP size equal to 4. All the reference pictures are located earlier than the current frame. These two modes achieve better compression performance than AI mode and do not suffer from the delay that RA mode introduces. The coding structure of the Low Delay P (LP) and Low Delay B (LB) modes shown in Figs. 3 and 4, respectively, are very similar.

Once we have presented all available coding modes in HEVC, we will show the different parallel techniques that can be applied.

Please cite this article as: H. Migallón et al., Synchronous and asynchronous HEVC parallel encoder versions based on a GOP approach, Advances in Engineering Software (2016), http://dx.doi.org/10.1016/j.advengsoft.2016.01.020