

A Parallel Implementation of JPEG2000 Encoder on Multi-GPU System

Bumho Kim, Jeong-Woo Lee, Ki-Song Yoon

ETRI(Electronics and Telecommunications Research Institute), Daejeon, Korea

{mots, jeongwoo, ksyoon}@etri.re.kr

Abstract— There has been an increase in the demand for a high-quality video codec that supports 4K (3,840 x 2,160) or more. JPEG2000 is an important technique for data compression, which has been successfully used in digital cinema and medical application. To process the high workload of JPEG2000 coding for large-scale video data, hybrid CPU/GPU platform is used to obtain high computing power. In this paper, we develop and implementation of the JPEG2000 compression standard in hybrid CPU/GPU platforms. Specifically, we develop multi-GPU implementations of the JPEG2000 encode to obtain high computing power and balance the load between cores and GPUs in the hybrid architecture. In our experiments with multi-GPU, we couple the JPEG2000 codec optimized for multicores and multi-GPUs and achieve high performance of the JPEG2000 compression.

Keywords— JPEG2000, Parallel System, Hybrid Platform, GPGPU, Digital Cinema

I. INTRODUCTION

There has been an increase in the demand for a high-quality video codec that supports 4K (3,840 x 2,160) or more. JPEG2000 is an important technique for data compression, which has been successfully used in the context of digital cinema and medical application that require high-resolution images. Due to the increasing spatial resolution of digital cinema and medical images, fast compression of image data is becoming an important and challenging objective.

JPEG2000 provides a practical set of features and offers numerous advantages over the previous standards [1]. JPEG2000 standard is based on wavelet technology and a layered file format that offer flexible lossy-to-lossless compression, irreversible compression that preserve image accuracy, and advanced functionality of image data management systems.

The JPEG2000 codec, which shows a higher compression and enables higher resolutions, entails a much more complex process with an enormous amount of computations. For 4K and 8K video containing large amounts of data, JPEG2000 encoders have a very high CPU demand, and it is hard for a single core computer to deal with such complex coding computations [2, 3].

To process the high workload of JPEG2000 coding for large-scale video data, hybrid CPU/GPU platform is used to obtain high computing power. During recent years, multimedia software has been ported to multi-core and GPU

architecture. Transition to hybrid CPU/GPU platforms in high performance computing is challenging in the aspect of efficient utilization of the heterogeneous hardware and existing optimized software [4].

In this paper, we develop the implementation of the JPEG2000 compression standard in hybrid CPU/GPU platforms. Specifically, we develop multi-GPU implementations of the JPEG2000 encode to obtain high computing power and balance the load between cores and GPUs in the hybrid architecture. In our experiments with multi-GPU, we couple the JPEG2000 codec optimized for multicores and multi-GPUs and achieve high performance of the JPEG2000 compression.

The rest of the paper is organized as follows. Section 2 and section 3 address JPEG2000 and GPGPU respectively. In section 4, we propose our scheme and define the system model. The performance results of the proposed method are shown in section 5. Finally, we provide concluding remarks on our scheme in section 6.

II. JPEG2000 OVERVIEW

JPEG2000 is an image compression standard from Joint Photographic Experts Group (JPEG). JPEG2000 provides compression performance superior to the current standards but also advanced features demanded by today's emerging applications. The features include superior visual quality at very low bit-rates, wavelet-based compression, progressive image transmission by improving pixel accuracy and random access to the code stream [5]. The JPEG2000 also provides great scalability in both quality and resolution and can work in both lossy and lossless mode on very large images. With all of these features, JPEG2000 is an ideal image standard for both mobile applications as well as high-quality applications such as medical imaging and digital cinema.

To meet these needs, JPEG2000 adopts a number of contemporary digital signal processing methods including a discrete wavelet transform (DWT) and embedded block coding with optimized truncation (EBCOT). The process DWT is a sub-band transform which transforms images from the spatial domain to frequency domain. Therefore, DWT can efficiently exploit the spatial correlation between pixels in an image. EBCOT is a two-tiered coder: Tier-1 is responsible for bit plane coding (BPC) and context adaptive arithmetic encoding (AE); Tier-2 handles rate-distortion optimization

and bitstream layer formation [6]. Figure 1 shows simplified block diagram of compression system defined by JPEG2000 standard.

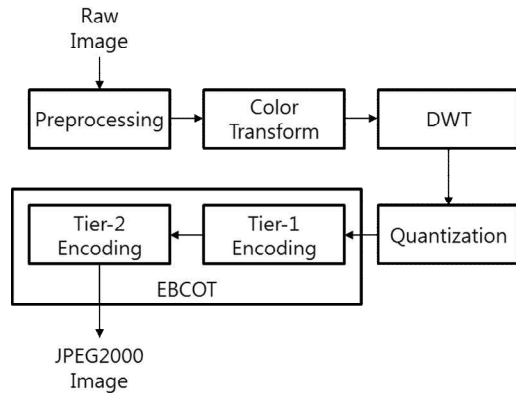


Figure 1. JPEG2000 Encoding Process

These advanced features and the superior compression performance yields higher computational demands which implies slower processing. Slow performance has long been noted as a major drawback of JPEG2000, particularly in software implementations. Resulting computational requirements of JPEG2000 are one of drawbacks hindering use of JPEG2000 in common application.

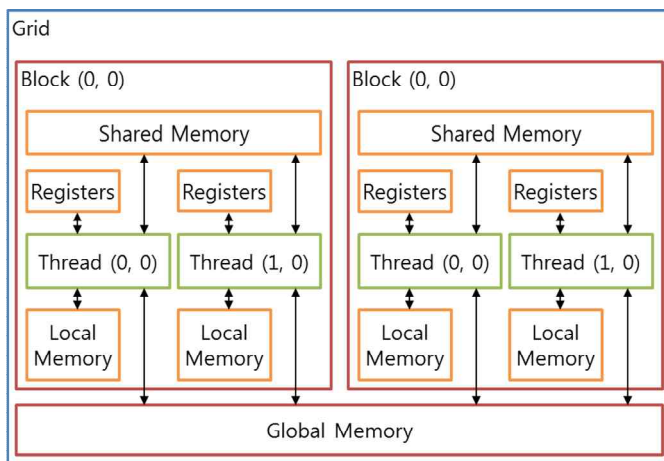


Figure 2. GPU Architecture

III. GPU OVERVIEW

Graphics processing units (GPUs) have become a popular computing architecture in recent years due to rapid increase of performance as compared to traditional CPUs [7].

CUDA is software and hardware platform designed for general purpose computing on GPUs in order to take full advantage of the maximum performance of GPUs in applications [8]. GPUs have a parallel architecture capable of running thousands of threads in parallel. In CUDA computing model, threads are grouped into thread blocks, and threads within thread block can cooperate among themselves using synchronization primitives by sharing data via a global memory and shared memory. The advantage of the global memory is that it can be accessed by all threads directly,

whereas the shared memory is only accessible to threads of one block. Compared to the global memory, the shared memory is considerably smaller and significantly faster. The data can be partitioned and fetched into the shared memory to provide higher throughput for more complex operations. While these architecture specifics of GPUs allow fine-grained parallelization for impressive increase of performance, it requires adaptation and re-formulation of algorithms resulting in more effective design on the GPU.

IV. SYSTEM MODEL

There has been a lot of effort to provide JPEG2000 applications with sufficient processing speed [9, 10]. We propose a parallel architecture of the JPEG2000 encoder in hybrid CPU/GPU platform to achieve scalability and a high encoding speed. To process the high workload of JPEG2000 encoding for large-scale video data, we develop the implementation of parallel encoder using multi-core CPU and multi GPUs.

Figure 3 shows a simplified block diagram of the JPEG2000 encoder to enhance the coding performance using multicore CPU and multi-GPUs platform. As mentioned in Sec. 2, the JPEG2000 encoder consists of several steps that are performed in consecutive order.

The first encoding step is component transform which converts the multiple color components data into another color representation. The component transform removes the inter-component redundancy that could be found in the image.

The next step is DWT which is a domain transform that transforms an image from spatial domain to frequency domain. This enables an intra-component special decorrelation that concentrates the image information in a small localized area. DWT can be performed by the lifting scheme based filter which has lower computational complexity and reduced memory compared to the former filter.

Once DWT is applied, all the resulting wavelet information is quantized, which means that wavelet coefficients are reduced in precision. The transformed coefficients are quantized using uniform scalar dead-zone quantization. The process of quantization introduces reduction of the data precision in order to achieve compression.

The encoding processes up to quantization are performed in multiple GPUs. In order to obtain high efficiency, the each component could be processed independently on separate GPUs. The first work flow is to copy image data from CPU RAM to global memory of GPU. Once image data is ready in global memory, the encoding process from color transform to quantization can be executed on GPUs.

After quantization, the integer wavelet coefficients still contain a lot of spatial redundancy. This redundancy is removed by context-based entropy coding (EBCOT) Tier-1 so the data is efficiently compressed into a minimum size bit-stream. The process of entropy coding is highly sequential and difficult to parallelize efficiently using many threads in GPU. Therefore, EBCOT step is performed in CPU. Each of these code-blocks is entropy coded separately, which gives potential for parallelization in multi-core CPU.

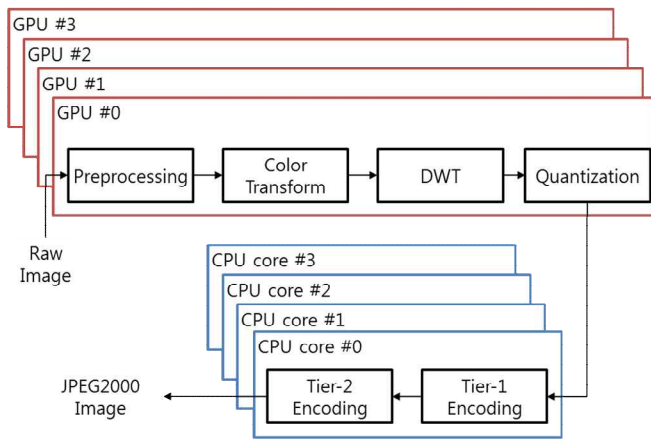


Figure 3. JPEG2000 Encoder Architecture on multi-core CPU and Multi-GPU System

The last step in encoding process is EBCOT Tier-2. This process is creating and ordering the packets for rate allocation. This basically consists of writing JPEG2000 bit-stream and creating the progression order. At the end of the computations all the data have to be saved on the CPU memory.

V. PERFORMANCE EVALUATION

In this section we present evaluation results of the proposed parallel implementation of JPEG2000 encoder. The proposed JPEG2000 encoder are implemented in the reference software, called “JasPer” [11], which is defined in Part 5 of the JPEG2000 standard. The Jasper encoder has been profiled to evaluate the proposed JPEG2000 encoder which is modified to be in parallel. The simulation platforms used in this experiment are shown in Tables 1. We used dual CPUs with an Intel Xeon w5590 at 3.33GHz clock frequency. For the GPU implementation, we used NVidia Geforce GTX 680 GPUs with CUDA as the development environment. Four sets of test sequences were selected, i.e., 2K (1920 x 1080, 2160 x 1080) and 4K (3840 x 2160, 4096 x 2160), as described in Table 2.

TABLE 1. EXPERIMENTAL SETUP

Options	Value
Processor	W5590 x 2
Cores	8 Cores
Clock frequency	3.33GHz
Operating System	Windows 7 64bit
Compiler	Visual C++ 9.0
GPU	NVidia GTX680 x 4
CUDA version	5.0

The experiment results for the multi-core implementation of the JPEG2000 encoder using a single GPU is provided in Fig. 4. The result shows that the parallel processing of multi-core CPU using parallel methods improves the overall performance. However, the parallel scheme in the multi-core CPU provides limited scalability, and cannot achieve a real-time response.

TABLE 2. TEST SEQUENCES

Class	Resolution	Frames
2K	1920 x 1080	480
2K	2048 x 1080	480
4K	3840 x 2160	480
4K	4096 x 2160	480

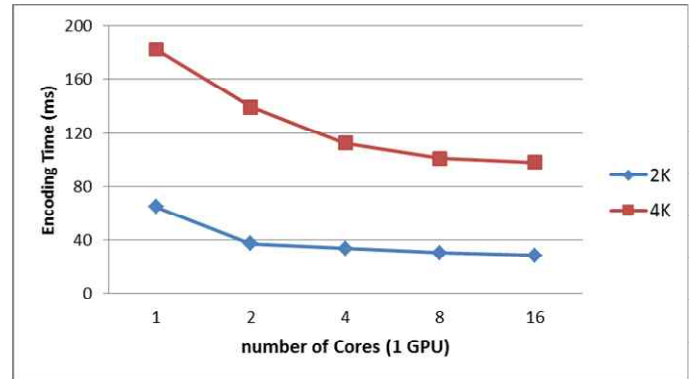


Figure 4. Encoding time of the proposed scheme (1 GPU).

Figure 5 shows the total encoding time of multi-GPU parallel processing for each resolution. This experiment shows the efficiency of the proposed multi-GPU JPEG2000 encoder, which use 16 cores CPU.

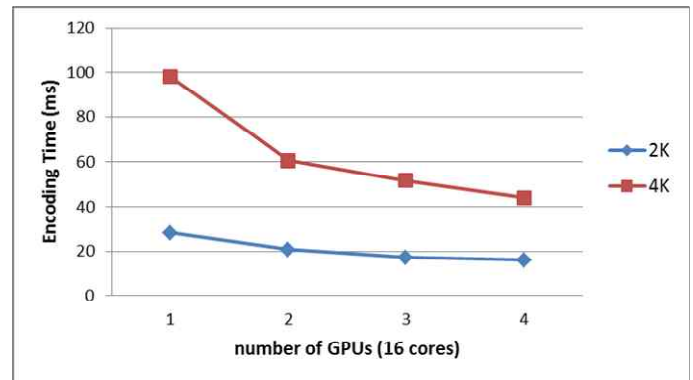


Figure 5. Encoding time of the proposed scheme (16 cores CPUs).

Table 3 shows the time spent on the encoding of the 1 frame for the test sequence encoding. This experiment shows comparison of the Jasper software and the proposed scheme for each resolution. The proposed parallel implementation using multi-CPU and multi-GPU can encode a 4K image within 50ms, and is about 100-times faster than the result of the Jasper software.

TABLE 3. ENCODING TIME OF JASPER AND THE PROPOSED SCHEME (MS)

Resolution	Jasper	Proposed Schem
2K(1920 x 1080)	0.7357	0.0167
2K (2048 x 1080)	0.8486	0.0159
4K (3840 x 2160)	5.6101	0.0485
4K (4096 x 2160)	4.9638	0.0440

These results show that a parallel processing using multi GPU provides significant gains in encoding time compared with the reference software. The statistics show that the parallel coding speed of the proposed scheme increases when the number of cores increases.

VI. CONCLUSIONS

We proposed a parallel architecture of the JPEG2000 encoder in hybrid CPU/GPU platform to achieve scalability and a high encoding speed. To process the high workload of JPEG2000 encoding for large-scale video data, we developed the implementation of parallel encoder using multi-core CPU and multi GPUs.

The simulation results verify that the proposed encoder is very suitable for highly complex video coding that involves a large amount of computations. Particularly for high-resolution video, i.e., 4K and 8K video that contain large amounts of video data, the proposed scheme can reduce the large encoding time and significantly improve the coding efficiency. The proposed architecture can be applied to implement a JPEG2000 decoder for high-resolution video sequences. As future work, we plan to implement an additional parallelism level by introducing SIMD and OpenMP to achieve real-time encoding.

ACKNOWLEDGMENT

This work was supported by MSIP (Ministry of Science, ICT and Future Planning) (10041539 High Compression, Low Loss Content Creation /Distribution /Display Technology Development for 8K-Video Service).

REFERENCES

- [1] ISO/IEC 15 444-1: *Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System*, 2000.
- [2] M. Ciznicki, K. Kurowski, and A. Plaza, "Graphics Processing Unit Implementation of JPEG2000 for Hyperspectral Image Compression," *Journal of Applied Remote Sensing*, vol. 6, 2012.
- [3] Le, Roto, Iris R. Bahar, and Joseph L. Mundy. "A Novel Parallel Tier-1 Coder for JPEG2000 using GPUs," *Application Specific Processors (SASP), 2011 IEEE 9th Symposium on. IEEE*, 2011.
- [4] Le, Roto, Joseph L. Mundy, and R. Bahar. "High Performance Parallel JPEG2000 Streaming Decoder Using GPGPU-CPU Heterogeneous System," *Application-Specific Systems, Architectures and Processors (ASAP), 2012 IEEE 23rd International Conference on. IEEE*, 2012.

- [5] J. Franco, G. Bernabe, J. Fernandez, *et al.*, "A Parallel Implementation of the 2D Wavelet Transform Using CUDA," *Parallel Distributed and Network-based Processing*, 2009.
- [6] K. F. Chen, C. J. Lian, H. H. Chen, *et al.*, "Analysis and architecture design of EBCOT for JPEG2000," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, May 2001.
- [7] Ciżnicki, Miłosz, *et al.* "Benchmarking JPEG 2000 implementations on modern CPU and GPU architectures." *Journal of Computational Science*, 2013.
- [8] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley, 2011.
- [9] J. Matela, V. Rusnak, and P. Holub, "GPU-Based Sample-Parallel Context Modeling for EBCOT in JPEG2000," *MEMICS*, 2010.
- [10] Matela, Jiří. "GPU-Based DWT acceleration for JPEG2000." *Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*. 2009.
- [11] M. D. Adams and F. Kossentini, "JasPer: A Software-Based JPEG-2000 Codec Implementation," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Oct. 2000.



Bumho Kim received the BS degree in computer science from Sogang University in 2000 and MS degree in information technology from Information Communication University in 2002, respectively. Currently, he is a senior researcher in the Creative Content Research Lab. at Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. His research interests include multimedia, video codec, digital cinema, and digital contents distribution.



Jeongwoo Lee received the B.S. degree in information and telecommunication engineering from Jeonbuk National University, Jeonju, Korea, in 1996, and the M.S. degree in information and communications engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 1998. He received the Ph.D. degree in the Information and Communications Department from GIST in 2003. He is currently working in Electronics and Telecommunications Research Institute (ETRI). His research interests include digital video coding algorithms, implementations for H.264 and HEVC, rate control algorithms for video coding, scalable video compression, and gpu-based coding algorithms.



Kisong Yoon received his M.S. and Ph.D degrees in Computer Science from New York City University in 1988 and 1993 respectively. From 1993, he was a principal member of Electronics and Telecommunications Research Institute (ETRI). His research interests are digital contents distribution, digital rights management and digital cinema/signage.