

# Optical Engineering

[SPIDigitalLibrary.org/oe](http://SPIDigitalLibrary.org/oe)

## **Parallel-pass architecture for embedded block coding with optimal truncation in JPEG 2000**

Goo-Rak Kwon  
Ramesh Kumar Lama  
Changjae Kim  
Jae-Young Pyun

# Parallel-pass architecture for embedded block coding with optimal truncation in JPEG 2000

Goo-Rak Kwon,<sup>a</sup> Ramesh Kumar Lama,<sup>a</sup> Changjae Kim,<sup>b</sup> and Jae-Young Pyun<sup>a</sup>

<sup>a</sup>Chosun University, Susuk-Dong, Dong-Ku, Gwangju, 501-759, Republic of Korea

<sup>b</sup>Sangmyung University, 31 Sangmyungdae-gil, Dongnam-gu, Cheonan, Chungnam 330-720, Republic of Korea

E-mail: [jypyun@chosun.ac.kr](mailto:jypyun@chosun.ac.kr)

**Abstract.** In this letter, we propose parallel-pass architecture for Embedded block coding with optimal truncation (EBCOT) entropy encoding in JPEG 2000. In the proposed method the time consuming sequential pass architecture is replaced with the parallel-pass approach. The experimental results show that the proposed method reduces the processing time by 22.6% as compared with the Taubman's Kakadu architecture of EBCOT. © 2012 Society of Photo-Optical Instrumentation Engineers (SPIE). [DOI: [10.1117/1.OE.51.7.070501](https://doi.org/10.1117/1.OE.51.7.070501)]

Subject terms: JPEG 2000; single pass coding; parallel pass coding.

Paper 120339L received Mar. 9, 2012; revised manuscript received Jun. 7, 2012; accepted for publication Jun. 7, 2012; published online Jun. 29, 2012.

## 1 Introduction

JPEG 2000<sup>1,2</sup> is the latest standard for still image coding. It has high compression performance and provides new features. However, the high computation complexity that grants excellent performance and rich features also restricts the real time applications of JPEG 2000.<sup>3,4</sup>

Embedded block coding with optimal truncation (EBCOT), proposed by Taubman, is the most complicated and time consuming part of JPEG 2000.<sup>5,6</sup> It is a bit-plane coder. Each bit-plane goes through three coding passes, called the significant propagation pass (Pass 1), the magnitude refinement pass (Pass 2) and the clean up pass (Pass 3). The context of a sample coefficient is formed according to the significant state of the sample and its eight neighbors within a  $3 \times 3$  context window. Next, the context data goes into the arithmetic coder. The scan order and the context window are shown in Fig. 1. During each pass, all the samples of the bit-plane are scanned to determine whether or not each sample is encoded in the current pass. Therefore, all the samples need to be scanned three times, requiring excessive processing time. Recently, a new method<sup>7</sup> was proposed by Jen et al. based on parallel processing by three passes. In this method, parallel processing of passes is achieved by scanning samples belonging to Pass 1 and Pass 2 concurrently and samples belonging to Pass 3 are delayed by one column. Since the Pass 1 and Pass 2 are executed concurrently, the output of Pass 1 cannot be used by Pass 2. We propose a fast context modeling method based on the parallel-pass

scheme. The strategy aims to process the three coding passes of the same bit-plane in parallel.

## 2 Proposed Method

In EBCOT, a proper coding pass for the sample must be determined first, then the sample is encoded during the coding pass. In this way, each sample in the bit-plane is encoded in one of the three passes. In order to reduce the processing time, three passes could be processed in parallel. However, the parallel processing causes a problem. If the three coding passes are concurrently executed, a sample in Pass 3 can become significant prior to its neighboring samples in Passes 1 and 2, resulting a wrong implementation of EBCOT. Moreover, in EBCOT, the processing results of samples in Pass 2 or 3 depend on those of Pass 1. However, in parallel-pass mode, samples in Pass 2 or 3 can not use the results of Pass 1.

In order to solve this problem, the coding operations for Passes 2 and 3 are delayed by one column to use the result of Pass 1, and Passes 2 and 3 are simultaneously processed. Figures 2 and 3 show the proposed scheme. The results of four samples (numbered as 1) are stored after they are encoded in Pass 1. Then, the samples (numbered as 2, 3) are encoded in Pass 2 or 3. In this case, the results of four samples (numbered as 1) are used as neighbors for Passes 2 and 3. After Passes 2 and 3 are completed, the two columns in box move to the right by one stripe. Hence in the proposed method, the time required to wait for the completion of scanning and coding of a strip for Pass 1 is reduced to the waiting time of single column. As a consequence, all three passes are encoded in one

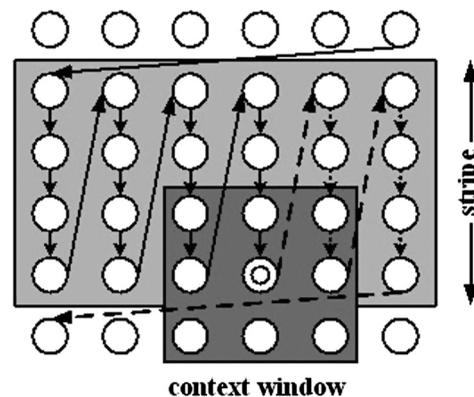


Fig. 1 Stripe oriented scan and context window concept.

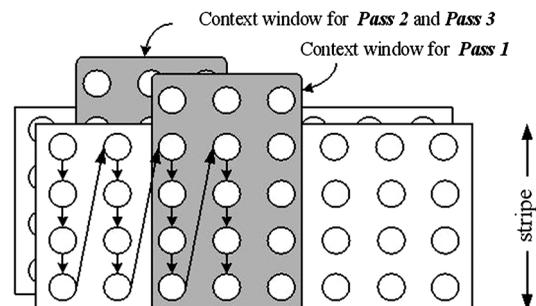


Fig. 2 Parallel processing of coding passes.

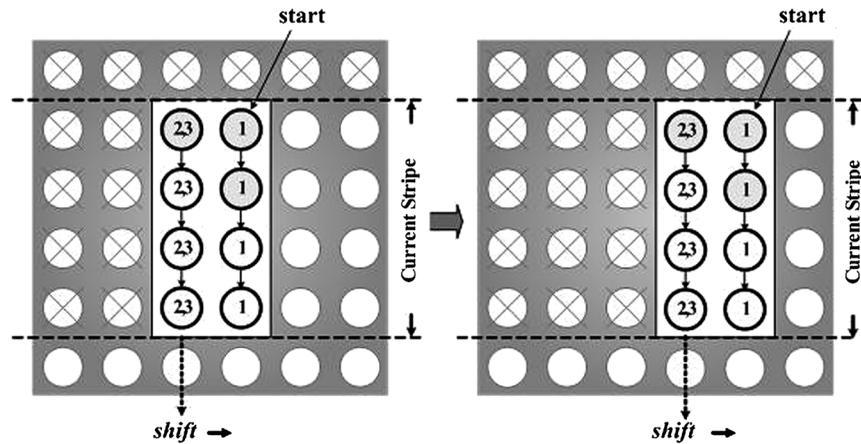


Fig. 3 Proposed parallel-pass architecture in detail.

Table 1 Experimental results for processing time of proposed architecture compared with David Taubman’s Kakadu, on three different images with size of 512 × 512.

	Kakadu (ms)			Parallel-pass (ms)			Pass-parallel/Kakadu			Total Time reduced (%)
	Pass 1	Pass 2	Pass 3	Pass 1	Pass 2	Pass 3	Pass 1	Pass 2	Pass 3	
Lena	297.8	140.3	522.8	298.6	88.5	357.4	1	0.63	0.68	22.5
Baboon	277.9	156.8	531.7	281.6	96.3	369.5	1	0.61	0.69	22.66
Peppers	269.7	157.2	533.9	272.8	92.6	378.3	1	0.59	0.71	22.6

scan. Additionally, Kakadu<sup>5,6</sup> uses the masking algorithm to extract a single bit-plane for each coding pass, and the three coding passes require three times of masking operations. In the proposed method, Passes 2 and 3 can reuse the result of the Pass 1, thus eliminating the masking overhead for Passes 2 and 3.

### 3 Results and Discussion

We tested the processing time of encoding three images (Lena, Baboon, and Peppers) to prove the effectiveness of the proposed method compared to the Taubman’s Kakadu architecture (version 3.4). Simulations have been conducted using a TMS3206416DSP. Test results are shown in Table 1. For Pass 1, the proposed method does not affect the execution time because there is no difference between the proposed method and the Taubman’s architecture. As shown in Fig. 3, all samples have to be scanned and samples associated with Pass 1 are encoded instantly. For Passes 2 and 3, the proposed method reduces the calculation time up to 41% (Pass 2) and 32% (Pass 3) and up to 22.6% of all three passes. This result indicates that the proposed method significantly reduces the processing time for scanning and masking. In average, the computation complexity of the whole EBCOT can be reduced by 22.6% as compared with the Taubman’s architecture. Since the proposed method changes only the scanning and coding time of the Passes in the original Kakadu method, the bit stream generated by the proposed method is same as that of original Kakadu method. Hence there is no change in PSNR performance of the proposed method and Kakadu method.

### 4 Conclusion

In this letter, we proposed a pass-parallel context modeling method to merge the three-pass coding into a single pass coding. With the processing of three coding passes concurrently the coding efficiency can be significantly improved. The experimental results show that the computational complexity is reduced by 22.6% as compared with Taubman’s architecture.

### Acknowledgments

This study was supported by research funds from Chosun University, 2011.

### References

1. M. Rabbani and R. Joshi, “An overview of the JPEG 2000 still image compression standard,” *Signal Process. Image Comm.* **17**(1), 3–48 (2002).
2. D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Massachusetts (2002).
3. A. N. Skodras, C. A. Christopoulos, and T. Ebrahimi, “JPEG 2000: the upcoming still image compression standard,” in *Proc. of the 11th Portuguese Conference on Pattern Recognition*, Porto, Portugal, pp. 359–366 (2000).
4. D. Santa-Cruz and T. Ebrahimi, “A study of JPEG 2000 still image coding versus other standards,” in *Proc. of the X European Signal Processing Conference*, Tampere, Finland, Vol. 2, pp. 673–676 (2000).
5. D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Trans. Image Process.* **9**(7), 1158–1170 (2000).
6. D. Taubman et al., “Embedded block coding in JPEG 2000,” in *Proc. of IEEE Int. Conf. Image Process.*, Vancouver, BC, Canada, Vol. 2, pp. 33–36 (2000).
7. J.-S. Chiang et al., “High efficiency EBCOT with parallel coding architecture for JPEG 2000,” *EURASIP J. Appl. Signal Process.* **2006**, 17–17 (2006).