High Efficiency Architecture of ESCOT with Pass Concurrent Context Modeling Scheme for Scalable Video Coding

Jen-Shiun Chiang, Ting-Hao Hwang, Tsung-Ta Lin, and Chih-Hsien Hsia

Department of Electrical Engineering

Tamkang University

Tamsui, Taipei, Taiwan

Email: chiang@ee.tku.edu.tw; talest@ee.tku.edu.tw;

Abstract—In this work, we propose a high efficiency hardware architecture of embedded sub-band coding with optimal truncation (ESCOT) with pass concurrent context modeling (PCCM) scheme for wavelet-based scalable video coding (SVC). PCCM can merge the three-pass process of bit-plane coding into a single pass process. It improves the efficiency of the ESCOT algorithm and reduces the frequencies of memory access, which can reduce the power consumption. Furthermore we use the parallel architecture scheme of PCCM to encode 4 samples concurrently, which improves the operation speed and can reduce 40% of internal memory requirement. We use Artison TSMC 0.18μm 1P6M standard cell library to design and implement the proposed concurrent context modeling. The simulation results indicate that PCCM can have an operation speedup of 9.5 compared to the standard context modeling of ESCOT, and it can operate for 1080p with frame rate of 30fps at clock rate of 125MHz.

I. INTRODUCTION

As the massive demand of video sequence in multimedia applications, the video sequence compression technique becomes more and more important. It does not only require high video quality and compression efficiency, but also needs more new functions to develop more applications. Scalable video coding (SVC) [1-2] is a novel and high efficiency coding technique and is expected as the next video sequence compression standard. It has better compression efficiency, superior video quality, error resilience, and enhanced functions than MPEG 2 and H.264/AVC [3]. The aim of SVC is to develop wide multimedia access services such that users can get multimedia information through variable devices from different locations and different platforms.

At present there are two major techniques for SVC, the DCT-based scheme [2] and the DWT-based scheme [1]. Although currently the main standard coding technique is the DCT-based scheme, the wavelet-based scheme has the advantage of resolution scalability, which the DCT-based SVC can hardly achieve [1][4]. The wavelet-based SVC uses the technique of embedded sub-band coding with optimal truncation (ESCOT) to encode the raw data on each bit-plane [1]. It is flexible because it encodes one sample on each bit-plane a time, and it can be extended to the object-based coding very easily. The Barbell lifting 3D wavelet coding [1] proposed by Microsoft Research Asia (MSRA) is one of the most effective coding scheme in the DWT-based SVC.

In the wavelet-based SVC coding scheme, the raw image sequence will be analyzed to a total of 40 different sub-bands after wavelet transformation and quantization [1]. Each of the 40 sub-bands is coded independently by entropy coding and then is exported as the encoded bit-streams. In the wavelet-based SVC the entropy coding takes a lot of computation efforts due to the encoding approach of ESCOT, which uses the fractional bit-plane coding technique. In ESCOT each sample on the bit-plane is scanned 3 times and encoded by only one of the 3 coding passes. Since it needs a lot of bit-level operations, it is very suitable to be implemented by hardware for real time applications.

Actually ESCOT is revised from the embedded block coding with optimized truncation (EBCOT) of JPEG 2000 standard [5]; the algorithms and some of the techniques of theses two approaches can be shared each other. However, the coding structure of ESCOT is different from EBCOT; ESCOT uses a new 3D context table for arithmetic coding that makes the algorithm to be more suitable for the SVC system [1]. Currently there are very few researches discussing about the hardware architecture and implementation of ESCOT. On the other hand, many papers have been published to analyze, design, and implement the EBCOT of JPEG 2000 [6-11]. Since the coding pass approach of the conventional EBCOT may reduce the encoding efficiency significantly, several researches try to modify the coding pass operation to increase the encoding efficiency [6-7]. Chiang et al. [6] proposed pass parallel context modeling (PPCM), to merge the three-pass coding scheme to a single pass coding scheme. PPCM can reduce 2/3 of the operation time in context modeling computation compared to the conventional approach. Besides the operation efficiency, PPCM can also reduce 20% of the internal memory requirement. However, in PPCM the order of the context modeling pattern (CX-D pair) is not generated

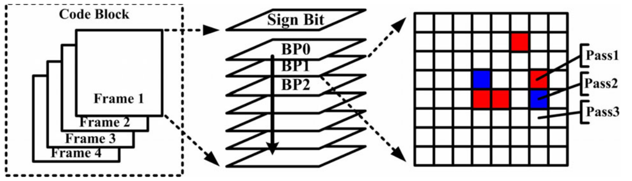

Figure 1. The data hierarchy for ESCOT.

following the order of the JPEG 2000 standard, and paper [6] proposed pass switch arithmetic encoder (PSAE) to finish the arithmetic encoding.

Like EBCOT, the context modeling of ESCOT needs 3 coding passes to encode a code sample, and it may waste the computation time and memory access time. In this paper, we propose a new context modeling algorithm and architecture called pass concurrent context modeling (PCCM) to increase the encoding efficiency and throughput of ESCOT. Like PPCM and CBCM [7] of EBCOT, PCCM merges the 3-pass coding to a single pass coding. Besides, PCCM encodes 4 samples concurrently to increase the context modeling operation speed further. The PCCM can have higher efficiency than that of CBCM. The requirement of the internal memory can be reduced about 40% compared to the conventional approach. Both operation efficiency and hardware cost of PCCM can be improved significantly.

II. CONTEXT MODELING AND PSAE FOR ESCOT

In the wavelet-based SVC, after the DWT transformation and quantization, the data in each sub-band are encoded by ESCOT. During the encoding process, it uses the orientation-invariant property of DWT to reduce the categories of the sub-band to simplify the encoding operation. Each sub-band is divided into several non-overlapped code blocks when processing ESCOT encoding. The code block is encoded by fractional bit-plane coding and arithmetic coding, and therefore each code block must be divided into one sign bit-plane and several magnitude bit-planes. ESCOT encodes the samples bit-plane by bit-plane. The data hierarchy for ESCOT is shown in Fig. 1.

A. *Context Modeling for ESCOT*

In ESCOT the encoding sequence is from the MSB bit-plane to the LSB bit-plane. The encoding sequence in each magnitude bit-plane is according to the significance, σ (significant state variable), of the samples and can be classified into three passes. Each sample in the magnitude bit-plane has its own σ. The initial value of σ is set to 0. During the encoding process in each bit-plane, if the value of the sample is 1, σ is set to 1 (significant). The significant state variable σ will become the initial value of the significant state variable σ for encoding the following bit-plane. After finishing encoding all the bit-planes, σ is reset to 0 for the next code block encoding. There are 3 kinds of coding methods in the context modeling of ESCOT, zero coding (ZC), sign coding (SC), and magnitude refinement (MR) coding, and the processing conditions of these 3 coding methods are described as follows:

- Zero coding (ZC): When σ = 0 regardless of the current sample is going to become significant or not, the current sample is coded by ZC. ZC also uses the significant information of the surrounding neighbors, as shown in Fig. 2, to encode the current sample.
- Sign coding (SC): When σ is changed from 0 to 1 in the current sample, it encodes the sign information by SC.
- Magnitude refinement (MR) coding: When σ = 1, it encodes the current sample by MR coding.

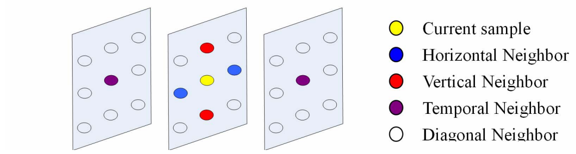

Figure 2. The surrounding neighbors of the current sample.

TABLE I. MEMORY SIZE FOR BIT-PLANE AND STATE VARIABLE

Category	Variable	Description	Size (bits)
Bit-Plane Data	Vp	Magnitude bit plane	4K×frame
	χ	Sign bit plane	4K×frame
Coding State Variable	σ	Significance state	4K×frame
	σ_2	Refinement state	4K×frame
	π	Bit-plane relationship variable	4K×frame

In the fractional bit-plane coding, each sample on the bit-plane should be scanned sequentially 3 times and encoded by only one of the three passes. The coding details of the three passes are described as follows:

- Significance Propagation Pass (or called Pass 1): Pass 1 processes the current sample that is not significant (σ = 0) and at least one of the surrounding neighbors (the relationship of the current sample and its surrounding neighbors is as shown in Fig. 2) is significant (σn = 1). It executes ZC coding first. If this sample becomes significant, it executes SC coding.
- Magnitude Refinement Pass (or called Pass 2): Pass 2 processes the current sample with σ = 1, and it executes MR coding.
- Normalization Pass (or called Pass 3): Pass 3 processes the current sample that is belonged to neither Pass 1 nor Pass 2. It executes the same coding processes as Pass 1.

In ESCOT it needs 5 blocks of memory to store bit-plane data and state variables for coding pass checking and encoding operations. Each block of memory takes 4K×(frame number) bits. The detail state variables and the corresponding memory size are shown in Table I. Vp and χ are used to record the current sample value and its sign. σ and σ2 record the significant state and first refinement state, respectively. π is used to record the sample that has been encoded by certain pass or not to prevent being re-encoded by another pass.

B. *Pass Switch Arithmetic Coding (PSAE)*

Arithmetic encoding (AE) is the core of entropy coding. It encodes the data according to the appearance probability of

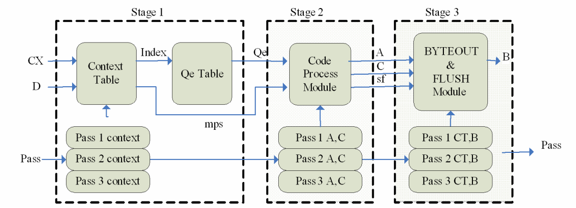

Figure 3. The block diagram of PSAE.

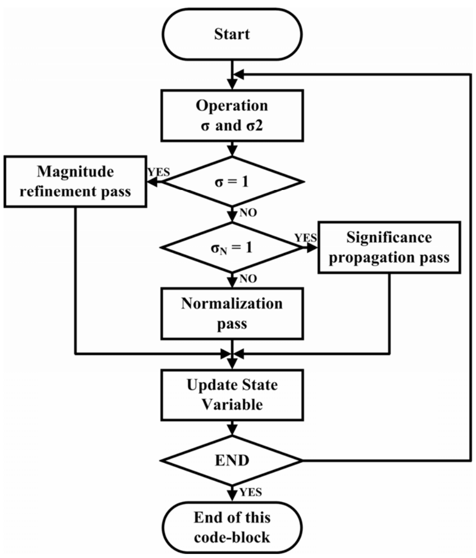

Figure 4. The coding decision flow.

the sample. The AE of ESCOT is the same as that of EBCOT, and here we adopt the PSAE proposed by Chiang *et al.* [6] for the proposed PCCM. The detail of PSAE can be referred to [6], and the block diagram of PSAE is shown in Fig. 3.

III. Pass Concurrent Context Modeling

ESCOT uses coding pass approach to encode the sample data. In the context modeling processing each sample is scanned 3 times, and the sample is encoded by its own pass and skips the other two passes. Since only 1/3 of the operations are valid, 2/3 of the operations and memory access times are wasted. In order to increase the operation efficiency and speed, we propose a new context modeling approach, pass concurrent context modeling (PCCM), to compute the context modeling. PCCM tries to merge the three coding pass operations to a single pass operation to improve the coding efficiency. Furthermore PCCM can process 4 samples in consecutive 4 frames concurrently to increase the operation speed.

A. *Coding pass merging*

The coding pass decision flow is shown in Fig. 4. In Fig. 4, σ_N indicates the significant state of the surrounding neighbors of the current sample. In order to allow the encoding sample not to wait till the finish of the operation of the surrounding neighbor samples (especially those samples have not been scanned but the belonged passes are less than the current sample), we must predict the significant states of the surrounding samples. Actually there are two kinds of predicted significant states. One is processed by Pass1 called virtual significant state, σ_v. The other is processed by Pass 3, and this kind of significant state can be calculated by logic OR of the current σ and Vp. The Pass 2 operation does not change the value of the significant state, and therefore we do not need to do prediction and the value of the significant state is equal to that of the virtual significant state.

For predicting σ_v we must use σ to decide whether the sample is belonged to Pass 1. If this sample is belonged to Pass 1 and Vp = 1, the significant sate of this sample is set to 1 in Pass 1. If the significant sate of this sample becomes 1 before the Pass 1 encoding, it is still significant after the Pass 1 encoding. We can use the following equation to find σ_v.

$$\sigma_v = (\text{the sample belonged to Pass 1 \& Vp}) \parallel \sigma \;;$$
$$= ((\sigma_N \,\&\, !\,\sigma) \,\&\, Vp) \parallel \sigma = (\sigma_N \,\&\, Vp) \parallel \sigma \qquad (1)$$

The neighbors of the current encoding sample can be classified into two categories, coded samples and un-coded samples, as shown in Fig. 5. The prediction of the significant states of the coded samples and un-coded samples is different. The significant state selection in different coding pass is shown in Table II.

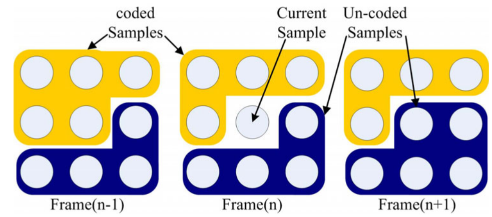

Figure 5. Coded samples and un-coded samples.

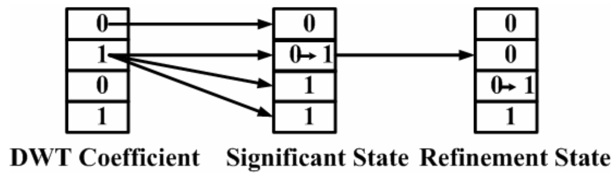

Figure 6. The relationship of Vp, σ, and σ2

TABLE II. The Significant State Selection in Coding Passes

usage	Significant State Selection		
	Coded samples	***Current samples***	***Un-coded samples***
σ_v Prediction	σ_v	σ	σ
Pass 1 coding	σ_v	σ	σ
Pass 2 coding	σ_v	σ_v	σ_v
Pass 3 coding	$\sigma_v \parallel V_P$	σ_v	σ_v

In ESCOT, σ and σ_2 are used to record the significant state and first refinement state, respectively. When analyzing Vp, σ, and σ_2, we find $\sigma_2(t)$ is equal to σ(t-1) and the relationship graph is shown in Fig. 6. σ(t) can be calculated from (2).

$$\sigma(t) = \sigma(t-1) \parallel Vp(t) = \sigma_2(t) \parallel Vp(t) \qquad (2)$$

Since σ can be calculated by σ_2 and Vp, we do not need to store the σ values and the memory for σ can be removed. Totally the PCCM architecture can save 40% of internal memory compared to the ESCOT standard

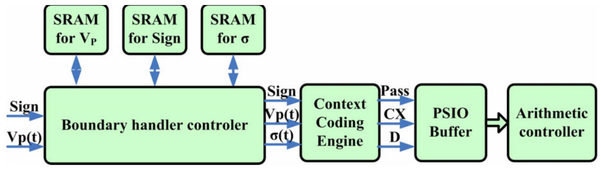

Figure 7. The ESCOT hardware block diagram

B. Frame sample parallel scheme

Due to the single pass operation of PCCM, each sample can be encoded independently each other. It is possible to encode several samples at the same time. In ESCOT sub-band encoding, the natural raster scan technique is used to scan several frames of the same code block. The scan order is as follows: temporal direction, horizontal direction, and vertical direction. The three-direction scanning may complicate the hardware implementation. However, if we can encode the sample at the same position of those frames, the hardware complexity may be reduced significantly. Because of the parallel characteristic of PCCM, we can encode the same position in 4 frames. This concurrent scheme does not only increase the computation speed, but also reduce the hardware complexity.

C. ESCOT architecture

Fig. 7 shows the hardware block diagram of ESCOT. Vp and Sign are inputs from the previous stage of SVC. ESCOT gets samples from the same position of the magnitude bit-planes in each frame at each clock cycle. When reading the first Vp with value of 1, the ESCOT reads sign data of that sample as well. After those data are processed by the boundary handler controller, the results are sent to the context coding engine to calculate CX-D pairs. Since the three coding passes are merged to a single one, the CX-D pairs are not generated following the original order: Pass 1, Pass 2, and Pass 3. The conventional arithmetic encoder (AE) of SVC cannot be used and the PSAE [6] is used instead. The PSAE architecture is shown in Fig. 3. Because the PCCM handles 4 samples concurrently, a parallel-in-serial-out (PSIO) buffer is needed in between PCCM and AE.

IV. Experimental Results and Comparisons

The proposed PCCM was designed and simulated by Artison standard cell library of TSMC 0.18μm 1P6M process. The circuit does not include the PSIO buffer and PSAE. This PCCM takes 95.75K gate counts and can operate at 125MHz.

We use standard QCIF as the testing source, and the testing data are processed by Microsoft Vidwav to finish DWT transform to generate the video sequence. Four video sequences, Akiyo, Foreman, Container, and Car phone, are tested in the proposed PCCM. Each video sequence consists of 30 frames that are extracted from the standard QCIF. The operating clock cycles of the four video sequences for both the standard context modeling and PCCM are shown in Table III. From Table III we find the PCCM can have an operation speedup of 9.5 compared to the standard context modeling. The qualities of both PCCM and standard context modeling are exactly the same.

TABLE III. The comparisons of the operating cycles .

QCIF	Standard	Proposed	Speed up
Akiyo	55,301,184 Clocks	5,884,646 Clocks	9.40x
Foreman	68,241,144 Clocks	7,091,642 Clocks	9.62x
Container	60,617,436 Clocks	6,415,904 Clocks	9.45x
Car phone	67,735,824 Clocks	7,052,744 Clocks	9.60x

V. Conclusion

We have proposed the PCCM architecture for ESCOT to increase the operation efficiency and speed of context modeling computation in this paper. In PCCM, the three coding pass operations are merged to a single pass operation. The single pass operation can both increase the coding efficiency and make the sample to be encoded independently each other. Therefore, we can encode more than one sample concurrently. In this work 4 samples are encoded concurrently. The PCCM can also reduce 40% of internal memory requirement. We use Artison standard cell library of TSMC 0.18μm 1P6M process to design and simulate the PCCM. The simulation results indicate that PCCM can reduce more than 90% operation time compared to the standard context modeling of ESCOT, and it can operate for 1080p with frame rate of 30fps at clock rate of 125MHz.

Acknowledgment

The authors would like to thank Dr. Yi-Shin Tung of GINM at National Taiwan University and Microsoft Research Asia for their generous support of Vidwav software for the video sequence verification to make this research possible.

References

[1] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, “Three-dimensional embedded subband coding with optimized truncation (3-D ESCOT),” Applied arid Computational Harmonic Analysis, vol. 10, pp. 290-315, May 2001.

[2] Text of ISO/IEC 14496-10: 2005/FDAM 3 Scalable Video Coding, Joint Video Team (JVT) of ISO-IEC MPEG & ITU-T VCEG, Lausanne, N9197, Sep. 2007.

[3] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Advanced Video Coding for Generic Audiovisual Services, July 2005

[4] J. Hua, Z. Xiong, and X. Wu, “High-performance 3-D embedded wavelet video (EWV) coding,” IEEE Workshop on Multimedia Signal Processing, pp. 569-574, Oct. 2001.

[5] ISO/IEC JTC1/SC29/WG1 15444, Information Technology- JPEG 2000 Image Coding System, Dec. 2000.

[6] J.-S. Chiang, Y.-S. Lin, and C.-Y. Hsieh, “Efficient pass-parallel architecture for EBCOT in JPEG2000,” IEEE Internation Symp. Circuits and Systems, vol. 1, no. 1, pp. 773-776, May 2002.

[7] J.-S. Chiang, C.-Y. Hsieh, J.-C. Liu, and C.-C. Chien, “Concurrent bit-plane coding architecture for EBCOT in JPEG2000,”IEEE Internation Symp. Circuits and Systems, pp. 4595-4598, May 2006.

[8] J.-S. Chiang, C.-H. Chang, C.-Y. Hsieh, and C.-H. Hsia, “High efficient EBCOT with parallel-coding architecture for JPEG2000,” EURASIP Journal on Applied Signal Processing, vol. 2006, pp. 1-14, 2006.

[9] H.-C. Fang, Y.-W. Chang, T.-C. Wang, C.-J. Lian, and L.-G. Chen, “Parallel Embedded Block Coding Architecture for JPEG 2000,” IEEE Trans. Circuits and Systems for Video Technology, vol. 15, Sep. 2005.

[10] C-C Chen, Y-W Chang, H-C Fang, and L-G Chen, “Analysis of scalable architecture for the embedded block coding in JPEG 2000,” in Proc. IEEE Int. Symposium on Circuits and Syst., ISCAS 2006, pp. 2609-2612, May 2006.

[11] Y. Li and M. Bayoumi, “A Three-Level Parallel High-Speed Low-Power Architecture for EBCOT of JPEG 2000,” IEEE Trans. Circhits Syst. Video Technol., no. 9, pp. 1153-1163, Sep 2006.