High-Performance Low-Complexity Bit-Plane Coding Scheme for MPEG-4 FGS

Hong-Yu Chao, Jia-Shung Wang, Juin-Long Lin, and Kai-Chao Yang
Department of Computer Science
National Tsing Hua University
Hsinchu, 300, Taiwan, R.O.C.
jllin@vc.cs.nthu.edu.tw

Chien-Ming Wu, Chun-Ming Huang, and Lan-Da Van
Chip Implementation Center (CIC)
National Applied Research Laboratories
Hsinchu, 300, Taiwan, R.O.C.
{wucm, cmhuang, ldvan}@cic.org.tw

ABSTRACT

MPEG-4 FGS (Fine Granularity Scalability) has received tremendous attentions because it has ability to adapt to the network bandwidth variation. In this paper, we present a novel and effective bit-plane coding technique to further improve the coding efficiency of MPEG-4 FGS. The proposed approach reveals three superiorities to the MPEG-4 FGS based bit-plane variable-length coding (VLC): (1) better video quality in about 1.2 dB in terms of average PSNR, (2) less memory requirement, and (3) lower implementation complexity and power dissipation. Thus, it is well suitable for efficient hardware or software implementations.

1. INTRODUCTION

In emerging communication applications, customized multimedia contents are required to stream to heterogeneous users over various network bandwidths. However, in a real situation, the transmission network has the bandwidth limitation, so the transmission quality is not guaranteed. To overcome this bottleneck, MPEG-4 FGS (Fine Granularity Scalability) has been proposed for adapting the variation of bandwidth efficiently [1-2]. MPEG-4 FGS encodes video sequences into two bitstreams: a non-scalable motion-compensated base layer and a fine-granular-scalable enhancement layer. The decoder can truncate the received bitstreams at any location of the enhancement layer, which provides fine granularity of reconstructed video quality proportional.

In the original FGS coding scheme [1-2], each bit-plane of the enhancement layer is coded by scanning blocks in the top-to-bottom and left-to-right order. Bits of each bit-plane are then encoded by run-length and variable-length coding (also called Huffman coding) with VLC (Huffman) tables. This coding strategy incurs two severe problems: (1) the coding events with a probability greater than 0.5 cannot be efficiently represented [3, 4]. (2) In practice, it is difficult to implement a high speed and/or low area complexity VLC encoder/decoder [5]. Those inspire many researchers to improve the FGS coding efficiency in many ways such as in [3, 4]. In [3], authors propose a way to improve coding efficiency, but the method suffers from the drawback of sophisticated VLC operation. To overcome this drawback, the context-based adaptive binary arithmetic coding (CABAC) is presented by greatly increasing the computational complexity [4].

The motivation behind our research is to develop a bit-plane coding scheme, named *adaptive bit-plane coding*, for efficiently encoding the MPEG-4 FGS enhancement layer. To improve coding efficiency, we adopt three different compression modes to encode different bit-planes. Simulation results expose that our proposed approach can improve the coding efficiency in around 1.2 dB in terms of average PSNR. In addition, the proposed adaptive bit-plane coding scheme does not need any VLC tables. The hardware implementation can thus be derived in a simple way with regular controlling circuitry. This not only reduces the hardware complexity, but also further results in the small power consumption in circuit implementation. Compared to the original MPEG-4 FGS based bit-plane coding, our proposed approach exhibits significant improvement in terms of performance, area, and power.

This paper is organized as follows: Section 2 briefly reviews the basic concepts of the MPEG-4 FGS coding. Section 3 describes the proposed *adaptive bit-plane coding* scheme. The resulting performance evaluation and comparison are then given in Section 4. Finally, Section 5 concludes this work.

2. BIT-PLANE CODING SCHEME OF MPEG-4 FGS

The generic block diagram of MPEG-4 FGS encoder is shown in Fig. 1 [1-4, 6]. MPEG-4 FGS encodes video sequences into a base layer and the enhancement layer. The base layer portion is the same as non-scalable prediction-based single layer MPEG-4. The enhancement layer part is to encode the difference between the original image and reconstructed picture with the base layer using bit-plane coding of the DCT coefficients. The residual bit-plane in the DCT domain can optionally be weighted (shifted) by selective enhancement or frequency weighting. After all the (weighted) DCT residues of a frame are extracted, the encoder finds the maximum absolute value of all the (weighted) DCT residues and

0-7803-9332-5/05/$20.00 ©2005 IEEE

determines the minimum number of bits, L, needed to represent its value in binary format. Note that L is also the maximum number of bit-planes (residual levels) for the enhancement layer. After that, the coefficients in the block are arranged in zigzag scanning order, and each coefficient is represented by L binary symbols, '0' and '1'. For each bit-plane of an 8 × 8 DCT residues block, the (RUN, EOP) symbol is formed and encoded using variable length coding.

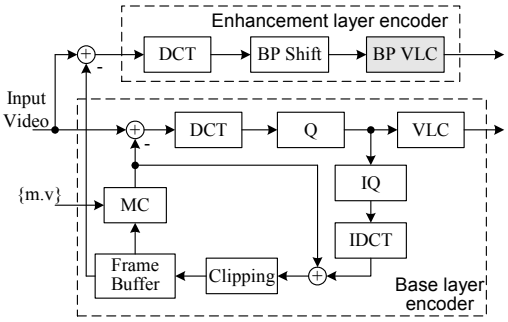

Fig. 1. The block diagram of MPEG-4 FGS encoder.

The first problem of this conventional run-length jointly by VLC coding scheme is that the coding events with a probability greater than 0.5 cannot be efficiently represented. This implies that the coding efficiency of original MPEG-4 FGS becomes extremely low in the lower bit-planes [3, 4]. Another problem is that MPEG-4 FGS encoder has multiple bit-plane VLC tables, which will take a larger chip area (memory size) and power (memory access power) in VLSI implementation [5]. In addition, it also has been mentioned in [6] that these VLC operations take more than half of the CPU time in the encoding process of the enhancement layer. In other words, the VLC is one of the critical performance, area, and power factors in MPEG-4 FGS. Consequently, it is important to have an efficient bit-plane coding scheme for improving the coding efficiency and reducing the VLC design complexity in MPEG-4 FGS enhancement layer.

This paper carries out a simple and effective coding scheme to improve the coding efficiency. Performance evaluation confirms that our approach exhibits significant performance advantages in comparison with the original VLC bit-plane coding under the same bandwidth. Moreover, our proposed scheme does not need any VLC tables, so it can be easily mapped onto a high-speed, low-complexity, and low-power circuit design. Our proposed scheme thus is a good coding scheme for a network with unpredictable bandwidth variation environments.

3. THE PROPOSED CODING SCHEME

3.1 Division into Three Coding Modes

In FGS, the enhancement layer is divided into bit-planes according to the DCT coefficients in binary representation. The occurrence of symbol '0' or '1' in each bit-plane is significantly distinct due to the characteristics of the DCT coefficients. While the bits distribution is uniformly distributed, the coding efficiency using a combination of run-length and VLC is low. We propose an *adaptive bit-plane coding* scheme to improve the entropy coding efficiency with low-complexity design in the enhancement layer. Before describing the proposed coding scheme, we explore the following properties:

Property (1): There are seldom coefficients containing the symbols '1' in the higher bit-planes. Thus, it needs to avoid processing these zero symbols in order to minimize bit-plane coding power.

Property (2): For the lower bit-planes, the symbol distribution of '1' and '0' is distributed uniformly; therefore, the coding efficiency of original MPEG-4 FGS is extremely poor in these lower planes.

Property (3): We find that the front coefficients of the middle bit-planes contain many symbols '1'. Nevertheless, there are some sparse symbols '1' in the rear coefficients of the middle bit-planes.

Since the occurrence probability of the symbol of '0' or '1' has different properties in different planes, we adopt three different compression modes to encode different planes according to above properties. Three compression modes are referred to as the "*bypass*", "*hybrid*", and "*coordination*", respectively, and describe as follows:

3.1.1 Coordination compression mode

To encode the bit-planes with sparse '1' efficiently, a novel compression methodology, named *Coordination*, is proposed. The procedures are as follows:

Step 1: Scan the 8 × 8 block in a left-to-right (column c = 0 to 7) and top-to-bottom (row r = 0 to 7) order.

Step 2: For r = 0 to 7, check the condition: Is any nonzero symbols (i.e. the symbol of '1') in this row?

Step 2.1: If the r-th row contains zero symbols only, a prefix '0' is used to denote that there exists no nonzero symbols in this row. Go to Step 3.

Step 2.2: If there are nonzero symbols, a prefix '1' is used to specify the existence of each nonzero symbol. Then, A-bits are used to indicate the location of this nonzero symbol. The value of A is determined as:

$A_i = \lceil \log_2(7 - P_{i-1}) \rceil$ for $1 \le i \le 8$ and $P_0 = 0$.

where $\lceil x \rceil$ denotes the greatest integer less or equal to x, i is the number of nonzero symbols in the row r, and P_i is the location of nonzero symbol i.

Step 2.3: After compressing the last nonzero symbol in the r-th row, a symbol '0' is used to notify the decoder that the rest symbols of this row are all zero symbols.

Step 3: End.

These steps are demonstrated in the example as depicted in Fig. 2. For the $r = 2$ of Fig. 2, the $P_1 = 3$, $P_2 = 5$, and $P_3 = 6$. Based on the equation in Step 2.2, we can deduce that $A_1 = \lceil \log_2(7 - P_0) \rceil = \lceil \log_2(7 - 0) \rceil = 3$-bits and the location of the first nonzero symbol is 3 = [011]. For the second nonzero symbol, the $A_2 = \lceil \log_2(7 - P_1) \rceil = \lceil \log_2(7 - 3) \rceil = 2$-bits and the location is noted as [01]. Finally, $A_3 = \lceil \log_2(7 - 5) \rceil = 1$-bit and the location is noted as [0].

r \ c	0	1	2	3	4	5	6	7	Compression
0	0	0	0	1	1	0	0	0	1[011]1[00]0
1	0	1	0	0	0	0	0	0	1[001]0
2	0	0	0	1	0	1	1	0	1[011]1[01]1[0]0
3	0	0	1	0	0	0	0	0	1[010]0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	1[111]
6	0	0	0	0	0	0	1	1	1[110]1
7	0	0	0	0	0	0	0	0	0

Fig. 2. *Coordination* example.

3.1.2 Bypass and hybrid compression modes

In *bypass* mode, all input bits are bypass (no compression) straightly to output. Based on the *property (2)*, we know that it is not necessary to compress the input data for the lower bit-planes. Thus, *bypass* compression mode is used for the lower bit-planes in our *adaptive bit-plane coding* scheme. For the middle bit-planes, the *hybrid* (either *bypass* or *coordination*) compression mode is adopted because of *property (3)*. We count the number of symbols '1', and then check whether the number is larger than the threshold value (T). *Bypass* compression mode is used if the number of symbols '1' is larger than the threshold; otherwise, the *coordination* compression mode is adopted. Note that the best threshold value is $T = 16$ by our statistics from a large number of experiments.

3.2 The Adaptive Bit-Plane Coding Scheme

Since different planes have different data distributions, we adopt different suitable modes to encode different planes. Based on the above descriptions, we summarize the procedures as shown in Fig. 3:

Step 1: Check the condition: Is the plane $l = 0$ or 1? If not, go to step 3.

Step 2: For the Plane-0 and Plane-1, *bypass compression* mode is chosen. Go to step 5.

Step 3: Scan the plane l ($l = 2, 3 \ldots L$-1) in a left-to-right order. And, every 64 symbols are rearranged as an 8×8 block.

Step 3.1: If the block is a zero block (i.e. all 64 symbols in this block are zero), no coding is required and we use a flag '0' to notify the decoder.

Step 3.2: Alternatively, a flag '1' is used to represent that this block is a nonzero block.

Step 4: For all nonzero block, the compression modes are adopted based on Table I:

Step 4.1: Since there are seldom coefficients containing the symbols '1' in the larger bit-planes (i.e. Plane-(L-2) and Plane-(L-1)), the *coordination* compression mode is adopted.

Step 4.2: Otherwise, the *hybrid* compression mode is adopted for the Plane-2 ~ Plane-(L-3).

Step 5: End.

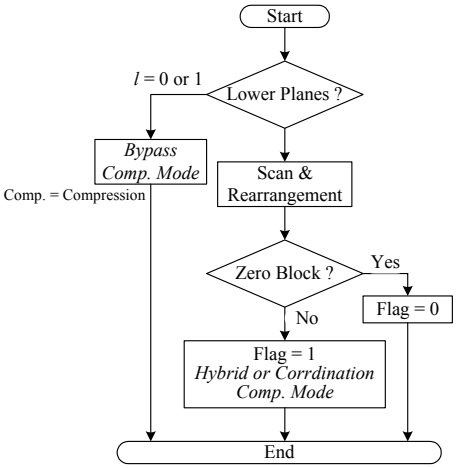

Fig. 3. Flow diagram of *adaptive bit-plane coding* scheme.

Table I: The coding mode of different planes.

Bit Plane (l)	*Compression Mode*
Plane-0 and Plane-1	Bypass
Plane-2 ~ Plane-(L-3)	Hybrid
Plane-(L-2) and Plane-(L-1)	Coordination

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

To demonstrate the effectiveness of our development, the computer simulation was performed. The test sequences, Claire, Coastguard, Salesman, Akiyo, and Foreman, in CIF format are selected in our experiment to cover a variety of video characteristics. Only the first frame is encoded as I frame and all the other frames are encoded as P frames. The quantization step of the base layer is 31, and the frame rate is 30 fps. In the experiment, the enhancement layer bitstream is truncated at the bit-rates from 256 kbps to 3072 kbps, with a unit of 256 kbps. In summary, Table II lists our experiment parameters.

Due to the pages limited, only two video sequences, Akiyo and Foreman, are demonstrated in this paper (see Fig. 4 and Fig. 5). As shown in Fig. 4 and Fig. 5, there is not too much performance gain at the lower bit-rate compared with MPEG-4 FGS. When bit-rate increases, the more significant coding gain can be obtained. On average, video quality is improved around 2 dB in Fig. 4 and around 1.1 dB in Fig. 5 over the MPEG-4 FGS in terms of PSNR. The average coding gain of the five test sequences (Claire, Coastguard, Salesman, Akiyo, and

Foreman) is around 1.2 dB.

Table II: Simulation environment.

Test sequences	Claire, Coastguard, Salesman, Akiyo, Foreman
Resolution	CIF
Frame rate	30-fps
Quantization step	31
Frame type	I frame and P frame
Enhancement layer bit-rates	256, 512, 768, 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072 kbps

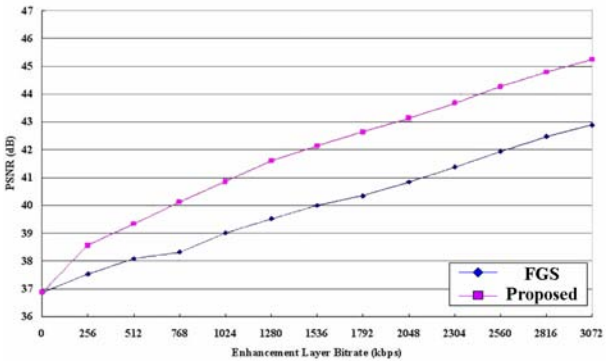

Fig. 4. PSNR performance of Akiyo sequence.

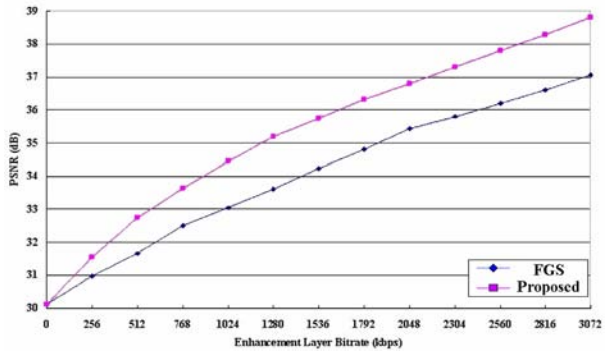

Fig. 5. PSNR performance of Foreman sequence.

The coding gain and design complexity of VLC are generally considered as the main issues in MPEG-4 FGS coding. The previous work [3] proposed a coding scheme that maintains the same picture quality as conventional FGS at low transmission and achieves about 0.2 dB better gains at a high transmission rate. The disadvantage is that it suffers from high complexity VLC design. For the context-based coding scheme, the VLC problem is solved [4]. However, it introduces more complicated arithmetic so as to degrade the merit. On the contrary, our proposed coding technique not only involves simple control operations but also doesn't possess any VLC tables. As listed in Table III, it is clear that the proposed *adaptive bit-plane coding* scheme is more area-efficient and power-efficient as well as has a better coding gain than its counterpart in [3, 4]. Those make our development a promising solution for the applications sensitive to the power and production cost.

Table III: Comparison of different FGS designs.

	FGS	[3]	[4]	Proposed
PSNR Performance	-	+0.2dB	N/A	+1.2dB
VLC Tables	yes	yes	no	no
Arithmetic Complexity	Normal	Normal	Much Higher	Normal

Throughout the comparisons, our proposed coding technique excels the best performance.

5. CONCLUSION

It is well known that memory size and power reductions are important for hardware design. In this paper, we propose a simple and effective bit-plane coding technique, which not only reduces significant area and power consumption but also improves the coding efficiency. In our coding scheme, each bit-plane is encoded using either *bypass* or *coordination* compression according to a smart selection mechanism. Due to its structural simplicity, the proposed coding technique can be easily mapped onto a high-speed, low-complexity, and low-power circuit design. Performance evaluation reveals that our development outperforms the MPEG-4 FGS based bit-plane coding by improving the PSNR in about 1.2 dB on average.

6. REFERENCES

[1] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301-317, March 2001.

[2] H. M. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia Streaming over IP," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53-68, March 2001.

[3] K. Matsuo, K. Takagi, A. Koike, and S. Matsumoto, "A bit-plane coding scheme of MPEG-4 FGS with high efficiency based on the distribution of significant coefficients," in Proc. IEEE Pacific-Rim Conf. Multimedia, Dec. 2002.

[4] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard, "*IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620-636, July 2003.

[5] S. H. Cho, T. Xanthopoulos, and A. P. Chandrakasan, "A low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 7, pp. 249 – 257, June 1999.

[6] Y. K. Chen and W. H. Peng, "Implementation of real-time MPEG-4 FGS encoder," in Proc. IEEE Pacific-Rim Conf. Multimedia, Dec. 2002.