A New Configuration of Adaptive Arithmetic Model for Video Coding with 3D SPIHT

Wai Chong Chia, Li-Minn Ang, and Kah Phooi Seng

***Abstract*— The 3D Set Partitioning In Hierarchical Trees (SPIHT) is a video coding algorithm extended from the SPIHT algorithm, which is initially introduced by A. Said and W. Pearlman for image coding. Previous works have show that the performance of 3D SPIHT with Arithmetic Coding (AC) is comparable to H.263 and MPEG-2. Moreover, the 3D SPIHT algorithm also supports progressive video transmission with controllable bit rates. This paper presents a new configuration of adaptive arithmetic model (also known as adaptive model) which can enhance the coding efficiency of AC, and achieves better performance in terms of Peak Signal-to-Noise Ratio (PSNR). The adaptive model is a very important module in AC, which is used to store the probability distribution of all the symbols that appear in a system. In the new configuration, each type of output bits in 3D SPIHT is assigned with a separate set of adaptive models. This new configuration takes into account the different probability patterns which exist in each type of output bits. On the other hand, the maximum frequency used to reset the adaptive model is also presented. This is a parameter that determines the adaptation rate of the AC. The simulation results show that the new configuration can improve the mean PSNR by 0.2 to 1.66 dB for various video test sequences in QCIF and SIF formats.**

***Index Terms*—Adaptive arithmetic model, arithmetic coding, SPIHT, video coding.**

I. Introduction

Due to the superior performance in image coding, many research works have been carried out to extend and improve the Set Partitioning In Hierarchical Trees (SPIHT) algorithm [1]. The 3D SPIHT algorithm is one of the extensions that attain remarkable performance in video coding. In terms of performance, it is comparable to H.263 and MPEG-2 without using any motion compensation. Moreover, it also supports progressive video transmission with controllable bit rates.

Several 3D SPIHT algorithms for video coding have been proposed in [2], [3] and [4]. In all these works, the Arithmetic Coding (AC) is adopted to further enhance the performance. However, the configuration of the adaptive arithmetic model (also known as adaptive model) for AC is not well reported in these works. This creates some difficulties in replicating the original algorithm as different persons can have different interpretation on the explanation. From our point of view, the configuration of the adaptive model adopted by the previous works is also not fully optimized. It will be shown later with our new configuration of adaptive model, which significantly improves the coding efficiency of AC for various video test sequences in QCIF and SIF formats.

The paper is organized as follows. A brief overview on SPIHT and 3D SPIHT algorithms is first given in section II. The proposed new configuration of the adaptive model is presented in section III. This is followed by the simulation results and discussion in section IV, and conclusion of the paper in section V.

Manuscript received September 26, 2008.

W. C. Chia, L-M. Ang, and K. P. Seng are with the School of Electrical & Electronic Engineering, The University of Nottingham Malaysia Campus, Jalan Broga, 43500, Semenyih, Selangor, Malaysia (phone: 603-89248000; fax: 603-89248002; correspond e-mail: [keyx7cwc / kezklma / kezkps] @nottingham.edu.my).

II. Background

A. SPIHT for Image Coding

Since the SPIHT algorithm is the basis of the 3D SPIHT, a brief overview is presented in this section. Moreover, the type of output bits in the coding process is clarified during the explanation of the algorithm. The clarification is very important in understanding the new configuration of the adaptive model.

Before using the SPIHT algorithm to encode an image, the 2D Discrete Wavelet Transform (DWT) is performed to decompose the image into multiple subbands. With the coefficients in the lowest frequency subband serving as the root, the SPIHT algorithm uses the Spatial Orientation Tree

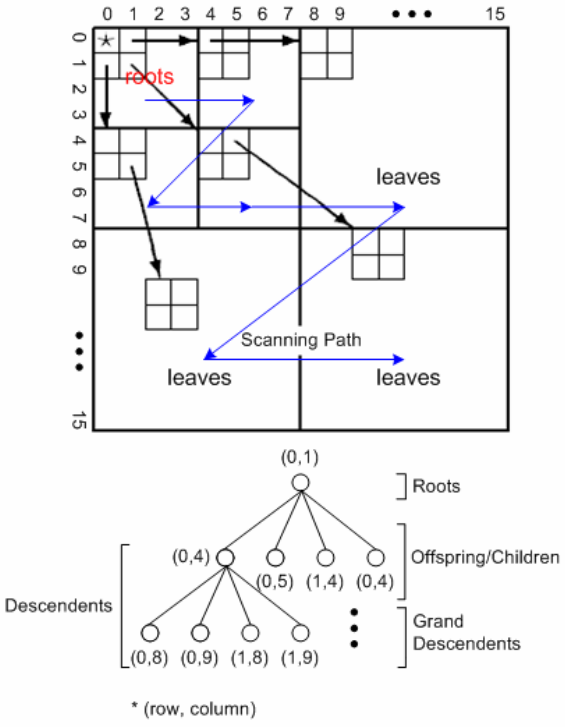


Figure 1. The 2D SOT structure adopted by SPIHT.

(SOT) structure to encode the wavelet coefficients. In this case, a node in the tree has either 4 offspring or no offspring. The offspring is formed by grouping the wavelet coefficients in 2 x 2 adjacent pixels. Fig. 1 illustrates the parent-offspring relationship of the SOT structure.

The SPIHT algorithm uses 3 lists which are called the List of Insignificant Pixels (LIP), List of Insignificant Sets (LIS), and List of Significant Pixels (LSP) to identify the status and control the coding process of wavelet coefficients. Initially, all the coordinates of coefficients located in the lowest frequency subband are added to the LIP. The entries in the LIS are similar to the entries in the LIP, except that the coordinates of coefficients denoted with "*" in Fig. 1 are excluded. In addition to this, all the entries in the LIP are marked as Type A entries. On the other hand, the LSP is initially left empty as it is used to store the coefficients which are tested to be significant.

The encoding process of SPIHT algorithm is divided into two phases which are called the sorting phase and refinement phase. In the sorting phase, each entry in the LIP is tested against a predefined threshold 2^n, where n is the level of significance. If the value of the entry is larger than the threshold 2^n, the entry is considered as significant at this level. In this case, a '1' bit and its sign bit are sent. Otherwise, it is considered as insignificant and only a '0' bit is sent. The entry which is tested to be significant is moved to the LSP before testing another entry. Throughout the paper, the bit which defines the significance of the entry in LIP is labeled as *LIP SIG* bit, and the sign bit is labeled as *LIP SIGN* bit.

On the other hand, the testing process of the entry in the LIS is slightly different from that in the LIP. In the case for the entry in the LIS, all the descendents of the entry are tested against the threshold 2^n. If any one of the descendents is larger than the threshold, the entry will be marked as significant. A '1' bit is sent for the case of significant, and a '0' bit is sent for the case of insignificant. This bit which defines the significance of the descendents will be labeled as *SDF* bit.

When the entry in the LIS is marked as significant, the 4 offspring of the entry will be tested in the same way as the entry in LIP. This means that a '1' bit and a sign bit are sent for offspring which is tested to be significant, and a '0' bit is sent for the opposite case. The coordinate of the significant offspring will be added to the end of the LSP whereas the insignificant offspring will be added to the end of the LIP. In this case, the bit which defines the significance of an offspring is labeled as *LIS SIG* bit, and the sign bit is label as *LIS SIGN* bit.

After the testing process of the 4 offspring, the entry of the LIS which is initially marked as Type A will be shifted to the back of the LIS and marked as Type B. When coding a Type B entry in the LIS, all the descendents excluding the 4 direct offspring will be tested against the threshold 2^n. If any one of the remaining descendents is larger than the threshold, the entry will be marked as significant. A '1' bit is sent for the case of significant and a '0' bit is sent for the case of insignificant. In addition to this, the coordinates of the 4 direct offspring of the Type B entry are added to the end of the LIS as new Type A entries. This bit which defines the significance of the Type B entry is labeled as *SGDF* bit throughout the paper.

In the refinement phase, the n bit of each entry in the LSP which appears in the previous pass is sent. This bit will be labeled as *REF* bit in the rest of the paper. In this case, no bit is sent during the first pass of the encoding process. After the refinement phase is completed, the n value is decremented for the next pass where the entire process described above is performed again. The decoding process of SPIHT algorithm is just the direct reverse of the encoding process.

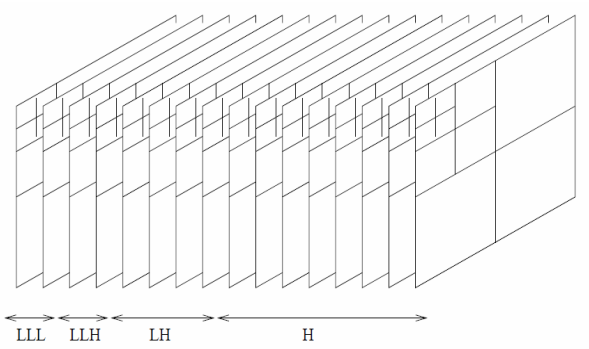
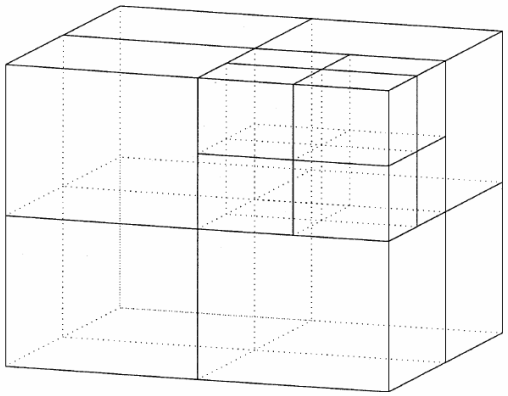


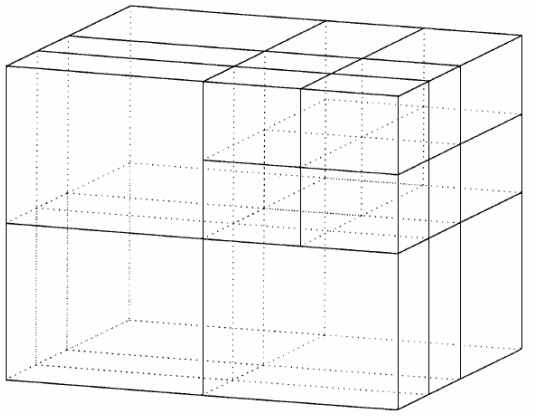
Figure 2. An example of performing the 3D DWT on a GOF with 16 frames. The 3D DWT uses 3 level of decomposition for both the spatial and temporal domains. (Source from [2])

B. 3D SPIHT for Video Coding

Generally, the encoding and decoding processes of the 3D SPIHT for video coding are totally same as the SPIHT for image coding. The main difference is the wavelet transform



(a)



(b)

Figure 3. The (a) symmetry and (b) decouple type of 3D DWT. (Source from [4])

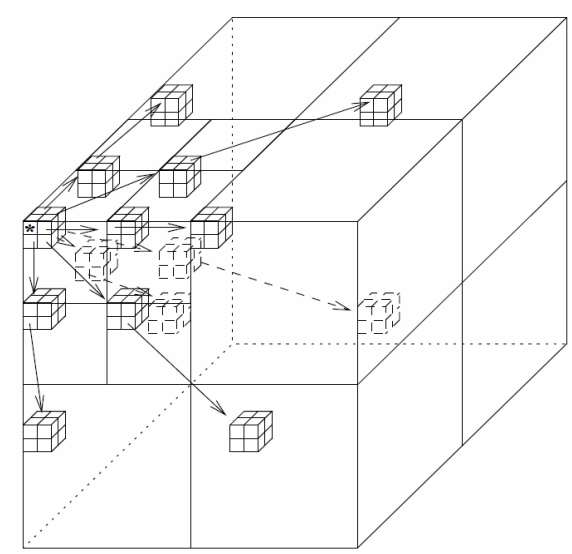


Figure 4. The symmetry 3D SOT structure. (Source from [2])

and the SOT structure that have been changed to 3D. These do not affect the clarification of the type of output bits which is stated previously.

An example of performing the 3D DWT on a Group of Frames (GOF) with 16 frames is shown in Fig. 2. First, a 2D DWT is performed on the spatial domain of each frame in the GOF. This is followed by a 1D DWT across the temporal domain of all the frames in the GOF. The number of subband created by the 3D DWT can be very different, depending on the configuration of its 2D DWT and 1D DWT that are applied to the spatial and temporal domain respectively.

Generally, there are two types of 3D DWT. One is called the symmetry 3D DWT as shown in Fig. 3(a), while another is called the decoupled or wavelet packet 3D DWT as shown in Fig. 3(b). The structure shown in Fig. 3 uses 2 decomposition levels for both the spatial and temporal domains. In this case, the numbers of subband in the symmetry 3D DWT and the decoupled 3D DWT are 15 and 21 respectively.

Since the data structure has changed from 2D to 3D, it is necessary to redefine the parent-offspring relationship of the SOT structure. The 3D SPIHT algorithm is not affected by the dimension, as long as the parent-offspring relationship of the tree structure is defined correctly.

Initially, the 3D SOT adopted in [6] is a direct extension of the 2D SOT. This 3D SOT is considered as a type of symmetry tree structure, since all the trees are symmetrically extended in both spatial and temporal domains. In the symmetry 3D SOT, a node is has either 8 offspring which are formed by a group of 2 x 2 x 2 adjacent pixels or no offspring (leaves). Similar to the 2D SOT, the nodes denoted with "*" have no offspring. The parent-offspring relationship of the symmetry 3D SOT is shown in Figure 4.

III. New Configuration of Adaptive Model

The AC introduced in [5] is adopted for our proposed configuration. This AC is also adopted by the previous works presented in [2], [3] and [4]. The main advantage of this AC is the isolation of adaptive model from the coding algorithm. Hence, the configuration can be modified easily.

Generally, the AC is separated into adaptive model and coding algorithm. The adaptive model is used to store the frequency or probability of the encoded symbols. It is updated after a symbol is encoded. The coding efficiency of AC is very much affected by the probability distribution in the adaptive model. When the frequency is skewed to certain symbols, the compression rate of AC will increase.

In 3D SPIHT, the information of the 8 offspring is usually encoded with AC as a single symbol to exploit the correlation. The amount of information to be encoded is depends on the number of insignificant pixels m in the group. In the previous works, several different adaptive models each with 2^m symbols, where $m \in \{1, 2, 3, 4, 5, 6, 7, 8\}$, are used to encode the information of the 8 offspring [2], [3]. From here, several questions may arise. Firstly, how many adaptive models were used and how they were configured. Secondly, each type of output bits is assigned with a separate set of adaptive models or the adaptive models were shared by few types of output bits. Thirdly, how were the sign bits encoded. Fourthly, what is the maximum frequency used to reset the adaptive models.

According to [1], the gain in coding the sign bits with AC is very little. It is true when the sign bits are view in terms of bit by bit. In this case, the numbers of positive and negative coefficients are close to each other. But from our observation, when the sign bits of the 8 offspring are coded as one symbol, certain patterns can occur more frequently than others. Due to this reason, it is possible to achieve some gain in coding the sign bits as a single symbol with AC in 3D SPIHT.

Other than the probability distribution, the maximum frequency also has significant effect on the coding efficiency of AC [5]. This maximum frequency will determines the adaptation rate of the AC. Furthermore, it also affects the amount of memory that is required to store the frequency for a symbol in the adaptive model. In our new configuration, the

Table 1. The proposed configuration of adaptive models for different type of output bits in 3D SPIHT.

Type of Output Bits	**Number of Adaptive Models with 2^m symbols**								**Number of Adaptive Models for Each Type of Output Bits**
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	
LIP SIG	1	1	1	1	1	1	1	1	8
LIP SIGN	1	1	1	1	1	1	1	1	8
SDF	1	1	1	1	1	1	1	1	8
LIS SIG	0	0	0	0	0	0	0	1	1
LIS SIGN	1	1	1	1	1	1	1	1	8
SGDF	1	0	0	0	0	0	0	0	1
REF	1	0	0	0	0	0	0	0	1
Total	6	4	4	4	4	4	4	5	43

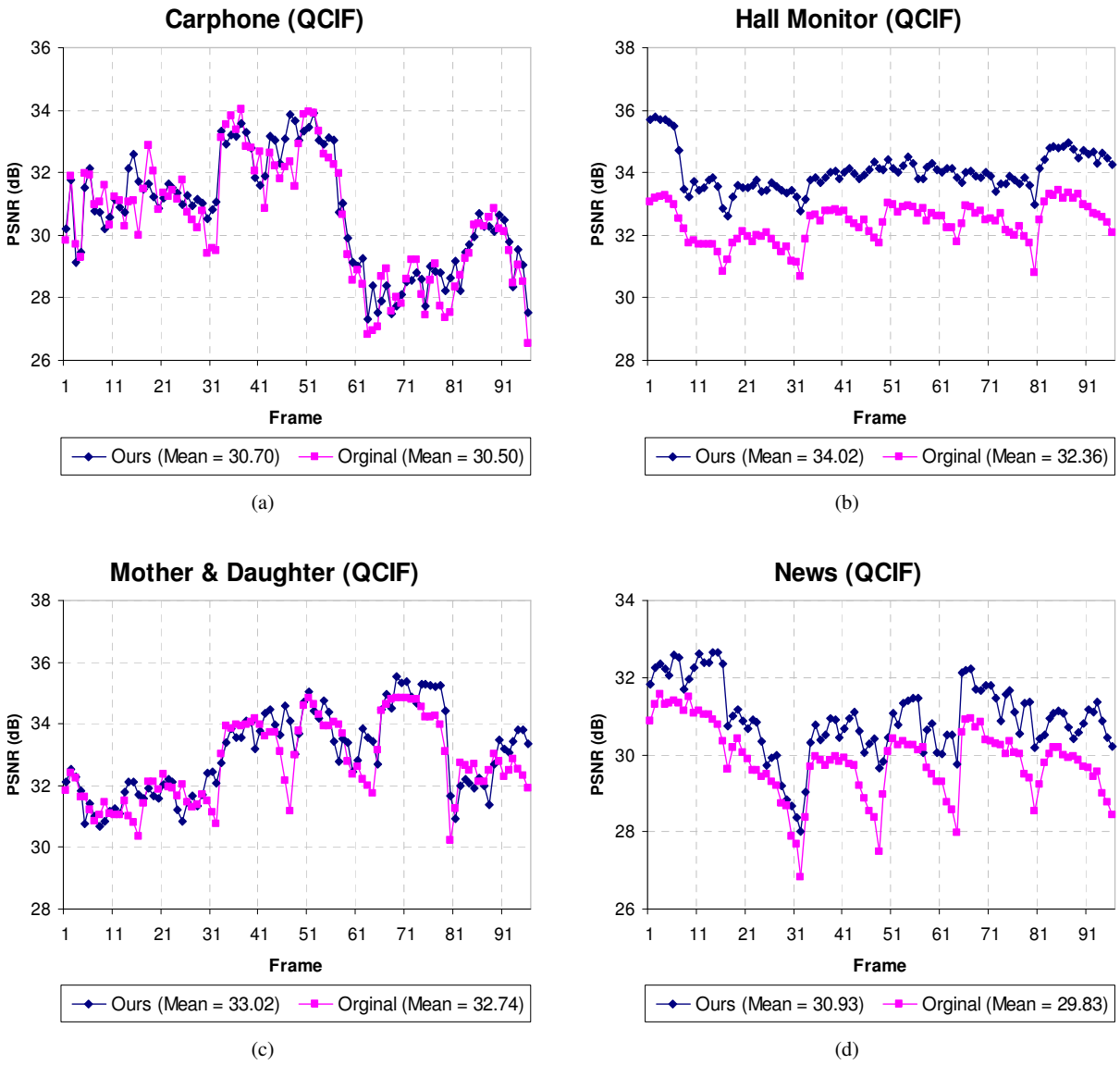


Figure 5. The simulation results for QCIF video sequences (a) Carphone, (b) Hall Monitor, (c) Mother & Daughter, and (d) News coded at 30 kbps with 10 fps.

maximum frequency is set to 64, which is different from the optimal value reported in [6]. This value is obtained from an analysis that will be explained in Section IV.

The proposed configuration of the adaptive models is summarized in Table 1. Even though the configuration of adaptive models for the *LIP SIG*, *LIP SIGN*, *SDF*, and *LIS SIGN* bits are the same, they are not sharing the same set of adaptive models. Each of them is assigned with a separate set of adaptive models with 2^m symbols, where $m \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. The adaptive model with different number of symbols is required to adapt for the case, where only the number of insignificant pixels m in the group are encoded. This is different for the *LIS SIG* bit because it always comes with 8 offspring at a time. Hence, only one adaptive model with 256 symbols is needed. On the other hand, the *SGDF* and *REF* bits are coded bit by bit separately with adaptive model of 2 symbols. Degradation in performance is observed during an attempt to code these bits in the same way as others.

IV. Simulation Results and Discussions

The proposed configuration is applied to the 3D SPIHT with AC and tested on various video sequences in QCIF and SIF formats. The decoupled 3D DWT and 3D SOT shown in Fig. 3(b) and Fig. 4 respectively are adopted in the simulation. For the 3D DWT, we used 3 level of decomposition for the spatial and temporal domains in QCIF video sequences with frame size of 176 x 144, and SIF video sequences with frame size of 352 x 240. This is different in SIF video sequences with frame size of 352 x 288. In this case, we used 4 level of decomposition for the spatial and temporal domains. The 9/7 tap [7] and Haar wavelet filter is adopted for the spatial domain and temporal domain respectively.

A. Simulation for QCIF Video Sequences

The simulation is performed with 10 fps by coding every third frame. Only the luminance component (Y) of frame 0 to

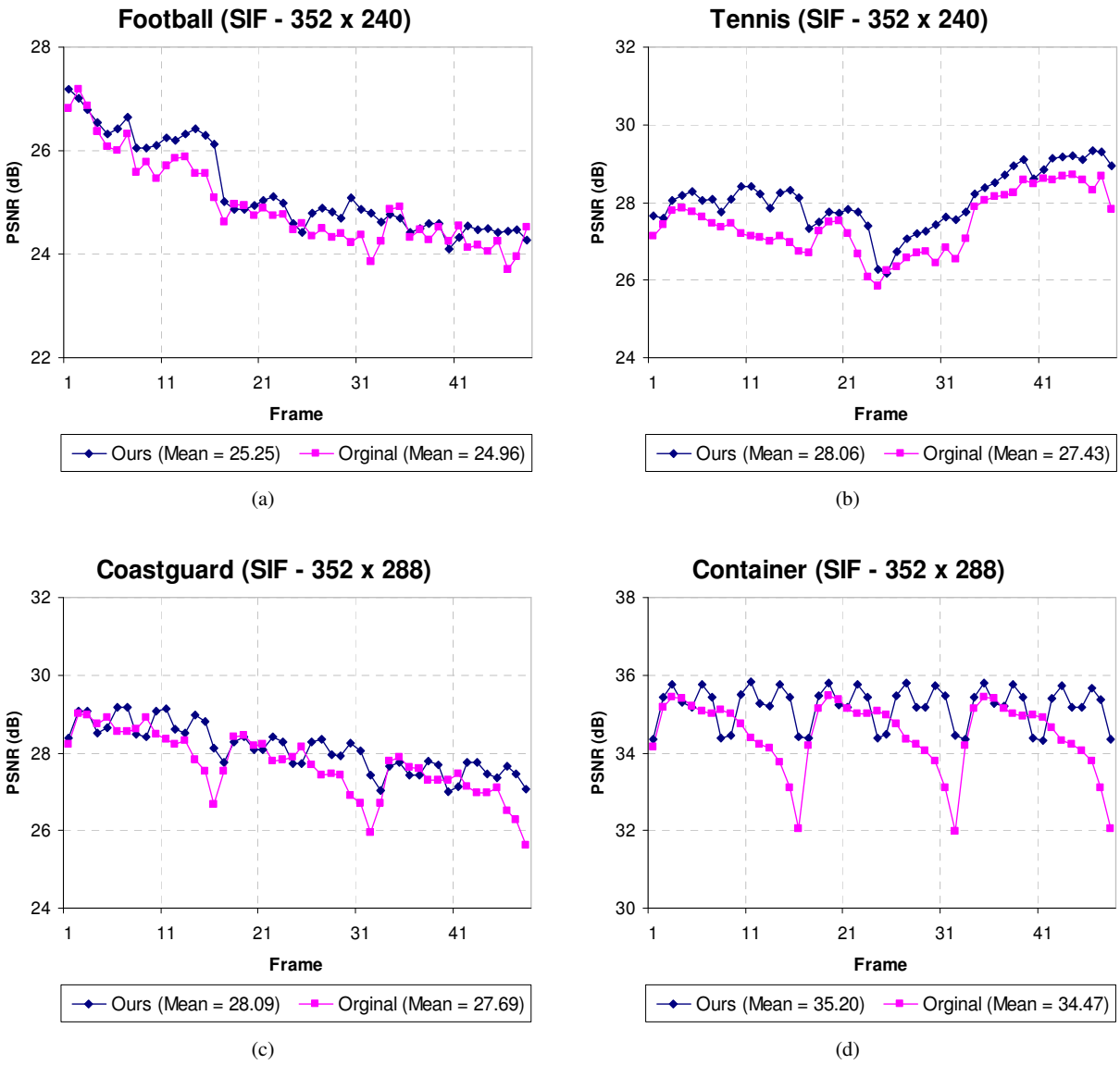


Figure 6. The simulation results for SIF video sequences (a) Football and (b) Tennis are coded at 300 kbps with 30 fps, whereas (c) Coastguard, and (d) Container sequence are coded at 360 kbps with 30 fps.

285 is considered in the simulation, and the GOF size of 16 frames is chosen. Since the simulation is performed by coding every third frame, the actual number of frames encoded is 96. The result of the 3D SPIHT with AC that uses the proposed configuration (Ours) is compared to the result generated from [8]. It should be noted that only the result for encoded frames are shown in Fig. 5.

B. Simulation for SIF Video Sequences

Unlike the setting applied to QCIF video sequences, the simulation for SIF video sequences is performed with 30 fps by coding every frame. Only the luminance component (Y) of frame 0 to 47 is considered. The GOF size remains unchanged as 16 frames. In this case, the total number of frames to be coded is 48. All the simulation results are shown in Fig. 6.

C. Adaptation of Adaptive Model

The mean PSNR achieved by the proposed configuration is 0.2 to 1.66 dB higher than the original configuration adopted by the 3D SPIHT. By using different sets of adaptive models for different types of output bits, better adaptation can be achieved as different types of output bits have different probability patterns. This also prevents the probability pattern of one type of output bits from affecting the others.

For example, assuming that there are 2 models with one of the probability patterns skewed to symbol *a* while another skewed to symbol *b*. If the 2 models are combined, the combined model may have 2 equal probable symbols when the frequency of symbol *a* and symbol *b* is very close to each other.

As stated previously, the maximum frequency also has significant effects on the compression rate. The number of output bits for a few video sequences simulated with different maximum frequency is shown in Table 2. By changing the maximum frequency to 64, few hundred bits can be saved in the later bit plane for both QCIF and SIF video sequences. Moreover, it also helps to reduce the memory required to store the frequency of each symbol in the adaptive model.

Table 2. Number of output bits for various QCIF and SIF video sequences under difference maximum frequency. The GOF is set to 16 frames.

Test Sequence	QCIF							
	Carphone				Hall Monitor			
Max Frequency	32	64	128	256	32	64	128	256
Bit Plane 1	639	639	639	639	585	585	585	585
Bit Plane 2	1370	1369	1369	1398	1320	1320	1320	1349
Bit Plane 3	2496	2534	2567	2653	2967	2977	2979	3048
Bit Plane 4	5654	5709	5758	5863	7026	7021	7035	7143
Bit Plane 5	12823	12879	12949	13065	15739	15680	15729	15849
Bit Plane 6	31525	31518	31627	31764	31444	31249	31421	31458
Bit Plane 7	74360	74248	74445	74696	57367	56906	57240	57188
Test Sequence	SIF							
	Football				Container			
Max Frequency	32	64	128	256	32	64	128	256
Bit Plane 1	756	680	649	637	532	532	532	532
Bit Plane 2	1606	1455	1391	1364	1194	1194	1194	1221
Bit Plane 3	3751	3538	3447	3405	2735	2753	2761	2831
Bit Plane 4	7943	7679	7575	7535	7116	7122	7169	7307
Bit Plane 5	18483	18206	18160	18200	16653	16643	16771	16918
Bit Plane 6	60322	60087	60184	60454	40111	39999	40294	40401
Bit Plane 7	192610	192236	192445	193379	90346	90104	90778	90908

D. Memory Requirement

The main drawback of the proposed configuration is the amount of memory that is required by the adaptive model, since different types of output bits are now assigned with a separate set of model. It is more than sufficient to use one byte of memory to store the frequency for one symbol, because the maximum frequency can only go up to 64. If only the memory required to store the frequency of all the symbols is taken into consideration, the memory requirement can be calculated by using (1), whereby $TNAM2^m$ represents the total number of adaptive models with 2^m symbols.

$$\text{Memory Requirement} = \sum_{m=1}^{m=8} TNAM\,2^m \times 2^m \times 1\,\text{Byte} \qquad (1)$$

In this case, the memory requirement for the proposed configuration is approximately 2.3 kB. For the case where different types of output bits are sharing the same set of adaptive model with *m* symbols, the memory requirement is only approximately 0.5 kB, depending on the setting of maximum frequency. But as compared to the amount of memory used to store the video frame or implement the 3D SPIHT algorithm [8], 2.3 kB of memory is almost negligible. Hence, the increment is not excessive.

V. Conclusions

The simulation results show that applying the proposed configuration of the adaptive models into 3D SPIHT with AC can significantly improve the performance by 0.2 to 1.66 dB. The main drawback is the increase in memory requirement introduced by the adaptive models. However, the increment is almost negligible as compared to the amount of memory required to store the video frame and implement the 3D SPIHT algorithm. By separating the adaptive models for each type of output bits, better adaptation can be achieved. This is due to the different probability patterns that exist in different type of output bits

References

[1] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, June 1996, pp. 243-250.

[2] B. J. Kim, Z. Xiong and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, December 2000, pp. 1374-1387.

[3] B. J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees," in *Proc. DCC*, Mar 1997, pp. 251-260.

[4] C. He, J. Dong, Y. F. Zheng, and Z. Gao, "Optimal 3-D coefficient tree structure for 3-D wavelet video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 10, October 2003, pp. 961-972.

[5] I. H. Witten, R. Neal and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, vol. 30, June 1987, pp. 520-540.

[6] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, December 1993, pp. 3445-3462.

[7] J. Chow and C. Lee, "Memory-efficient implementation of 3-dimensional zerotree video coding," in *Proc. Int. Conf. on Consumer Electronics*, 2002, pp. 350-351.

[8] B. J. Kim, W. A. Pearlman, and A. Said, "Three dimensional set partitioning in hierarchical trees compression," [Online] Available: http://www.cipr.rpi.edu/research/SPIHT/EW_Code/spiht3d_g_dos.zip (Last Accessed: 10 Jan 2009).

[9] M. Antonini, M. Barlaud, P. Mathiew, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, Apr 1992, pp. 205 – 220.