Low Rate Video Coding with Block Reordering in Wavelet Domain

Pao-Chi Chang[1] and Ta-Te Lu
Department of Electrical Engineering, National Central University, Chung-Li, Taiwan 320

ABSTRACT

In this paper, we present a novel energy compaction method, the selective block reordering, which is used with SPIHT (SBR-SPIHT) coding for low rate video coding to enhance the coding efficiency for motion-compensated residuals. The inter-frame coding basically includes three major parts - motion estimation, motion compensation, and motion-compensated residual coding. The motion estimation and overlapped block motion compensation (OBMC) methods of H.263 are used to reduce the temporal redundancy. The motion-compensated residuals are encoded in the wavelet domain. The block-mapping reorganization utilizes the wavelet zerotree relationship that jointly presents the wavelet coefficients from the lowest subband to high frequency subbands at the same spatial location, and allocates each wavelet tree with all descendents to form a wavelet block. The block reordering based on the threshold scan rearranges the significant blocks in the descending order of the energy. Then, the block reordering technique reorders the wavelet sub-blocks recrusively, according to the energy of each sub-block, to yield the maximum energy compaction that allows the SPIHT coding to operate efficiently on the motion-compensated residuals. Simulation results demonstrate that SBR-SPIHT outperforms H.263 by 1.28~0.69 dB on average for various video sequences at very low bit-rates, ranging from 48 to 10 kbps.

Keywords: SPIHT, H.263, wavelet video coding, block reordering.

1. INTRODUCTION

Very low bit-rate video coding is an interesting and challenging research field. Although the commonly used video compression standards, e.g. MPEG-4 [1] and H.263 [2], provide solutions for low bit-rate applications, they may generate annoying artifacts at very low rates. Namely, the discrete cosine transform (DCT) that is the main component in these video compression standards suffers from the blocking effect and mosquito noise at very low bit rates [3]-[6].

Embedded zero tree coding techniques such as Said and Pearlman's set partitioning in hierarchical trees (SPIHT) [7] and Shapiro's embedded zerotree wavelet (EZW) coding [8] exhibit the progressive coding capability. The embedding property of progressive coding allows the decoder to stop decoding at any point, without wasting any received information. The SPIHT coding is based on the pyramid structure to exploit the self-similarity across frequency bands, and the wavelet coefficients are organized into spatial orientation trees from the lowest subband to high frequency subbands. In image coding, SPIHT with a set partitioning sorting algorithm and an ordered bit plane transmission typically outperforms EZW and JPEG at a low bit-rate [9]. SPIHT has also been applied to video coding. Kim and Pearlman proposed 3D-SPIHT at the bit-rates of 30-60 kbits/s [10]. Lin and Gray proposed video residual coding based on SPIHT that applied Lagrangian optimization to the SPIHT encoder to achieve better rate-distortion performance than H.263 [11]. Although SPIHT coding is effective for images, however, it is still less efficient in representing block based motion-compensated residuals that have arbitrarily dispersed energy distributions in each prediction frame.

In this paper, SPIHT is applied to video coding with the selective block reordering technique to enhance the

[1] Correspondence: e-mail: pcchang@ee.ncu.edu.tw; phone: 886 3 4227151 ext. 4466; fax: 886 3 4255830.

coding efficiency for motion-compensated residuals. The motion estimation and motion compensation methods in H.263 are used to reduce temporal redundancy. The motion-compensated residuals are then encoded in the wavelet domain. The block-mapping reorganization technique utilizes the wavelet tree relationship that jointly presents the wavelet coefficients from the lowest subband to high frequency subbands at the same spatial location, and allocates each wavelet tree with all descendents to form a wavelet block. The block reordering based on the threshold scan rearranges the significant blocks in the descending order of the energy, to yield the maximum energy compaction that allows the SPIHT coding to operate efficiently on the motion-compensated residuals. Finally, motion vectors and the SBR-SPIHT coded bit-planes are encoded by the adaptive arithmetic coding [12].

The rest of this paper is organized as follows. Section 2 describes the selective block reordering structure while Section 3 describes the details of the motion-compensated residual coding. Section 4 presents the simulation results and conclusions are finally made in Section 5.

2. BLOCK REORDERING SPIHT VIDEO CODING STRUCTURE

The block diagram of SBR-SPIHT encoder scheme is illustrated in Fig. 1. The original SPIHT algorithm is used to encode the first intra-frame because SPIHT coding is one of the best methods to encode still images at a given target bit-rate. The inter-frame coding basically includes three major parts - motion estimation, motion compensation, and motion-compensated residual coding. The motion estimation and overlapped block motion compensation (OBMC) methods of H.263 that operate in the pixel domain are used to reduce the temporal redundancy. The motion-compensated residuals are first processed by the discrete wavelet transform (DWT) [13]. The block mapping allocates each wavelet tree with all descendents to form a wavelet block. Then, the block reordering relocates wavelet coefficients to yield maximum energy compaction and thus high efficient SPIHT coding. The details of the intra-frame coding and inter-frame coding are described separately as follows.

2.1 Intra-frame Coding

Intra-frames are encoded by the original SPIHT. The original SPIHT algorithm consists of two passes, namely, the sorting pass and the refinement pass. In the sorting pass, SPIHT sets a magnitude threshold, $T = 2^n$, where n is the level of significance. If any wavelet coefficient, $c_{i,j}$, satisfies $T \le |c_{i,j}| < 2T$, it is set as significant. The sorting pass is used to find the significant pixels that were defined as insignificant before the current threshold scan. The list of insignificant pixels (LIP), the list of significant pixels (LSP), and the list of insignificant sets (LIS) are employed to indicate whether the pixels are significant or not in the sorting pass. Each entry of the LIP produces one bit to describe its significance. If the pixel is significant, then it is moved to the LSP and a sign bit is transmitted. Otherwise, the pixel is insignificant and remains in the LIP. Next, each set in the LIS generates one bit to indicate if the existent descendents are significant or not, and the set is partitioned into subsets when the set has at least one significant descendent. In the refinement pass, all coefficients in the LSP that have been identified as significant in the previous sorting pass are refined bit by bit. The algorithm repeats the above procedure, and the level of significance n is decreased by one each time.

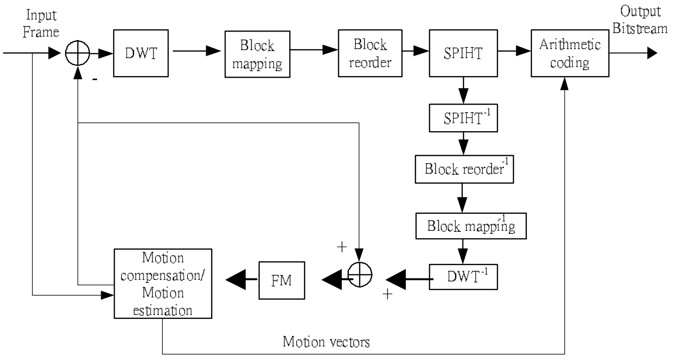


Fig. 1: The Encoder of SBR-SPIHT video coding system.

2.2 Inter-frame Coding

(1) Motion Estimation: The motion estimation we use in this work follows the H.263 standard, including OBMC and the possibility of four motion vectors per macroblock. The original frame is divided into non-overlapping 16×16 macroblocks with each macroblock containing four 8×8 blocks. For each macroblock, the one/four motion vectors are estimated to obtain the minimum distortion from the previous frame. The distortions are determined by the sum of absolute difference (SAD) in each macroblock. The motion estimation algorithm has intra/inter modes. If the motion estimation result is not accurate enough in this macroblock, then the intra mode is activated. Otherwise, this macroblock is set to be the inter mode and the motion vector is estimated for this macroblock.

(2) Motion Compensation: In SBR-SPIHT, OBMC of H.263 is used to reduce the overall energy of the motion-compensated residuals and the blocking effect at low bit-rates [3]-[4]. However, the blocking effect between inter/intra coded blocks is still serious because of different statistic properties. Therefore, in this work, the mean removing technique is used that subtracts the mean value of the intra block from the pixel values in an intra block. The mean value of an intra block is sent as side information.

(3) Motion-Compensated Residuals Coding: With the mean values removed, pixels in intra blocks and residual values in inter blocks are then combined to form a frame and coded with SBR-SPIHT algorithm in the wavelet domain. To improve the SPIHT coding efficiency, SBR-SPIHT utilizes the block mapping and the block reordering, which are detailed described in the next section.

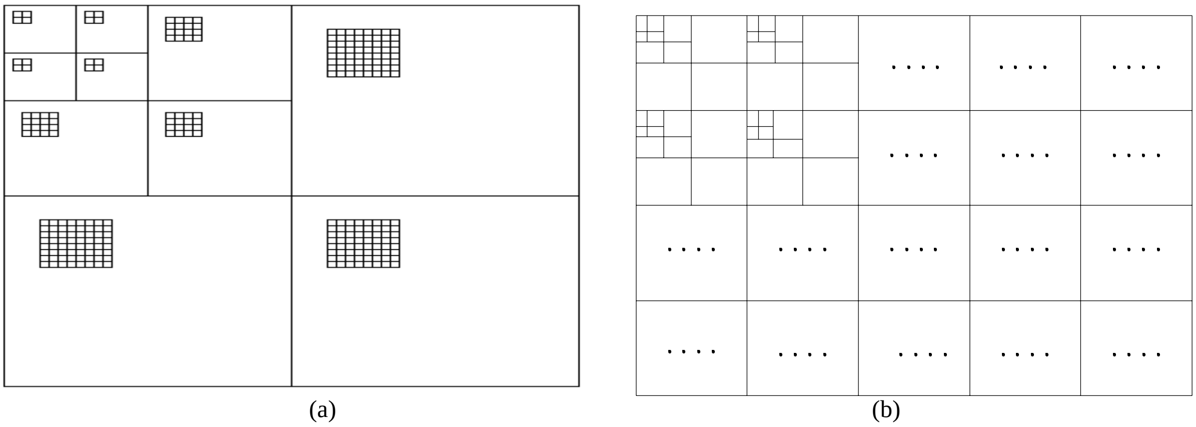
3. MOTION-COMPENSATED RESIDUALS CODING

The motion-compensated residuals are first decomposed into multiresolution subbands by L level discrete wavelet transform (DWT) with Daubechies-4 orthogonal filters [13]. The motion-compensated residuals have very different characteristics from natural images. The residuals usually have much more high frequency components that may not be efficiently coded by traditional image coding approaches. In other words, the DWT may decompose these residual components into large wavelet coefficients in high frequency bands without significant coefficients in the corresponding lower frequency bands. This property significantly reduces the coding efficiency of SPIHT because SPIHT coding needs to spend many redundant bits in the LIS and LIP finding those isolated significant coefficients in high frequency bands.

According to the SPIHT bits structure, all bits can be classified into three categories: sorting significance bits, sign bits, and refinement bits. The sorting significance bits are used to indicate the entry and descendents in the LIS and LIP that are significant or not. The LSP is used to transmit the sign bits and the refinement pass is used to refine the wavelet coefficients bit by bit. Sign bits and refinement bits represent the sign and magnitude of wavelet coefficients respectively and they are used to reconstruct the wavelet coefficients by bit planes. The block mapping and block reordering are described as follows.

3.1 Block-mapping Reorganization

The block mapping reorganizes the wavelet coefficients into wavelet blocks as shown in Fig. 2.



(a) (b)

Fig. 2: Block mapping (a) multiple-layer wavelet structure (b) reorganized wavelet blocks.

It utilizes the wavelet tree relationship that jointly presents the wavelet coefficients from the lowest subband to high frequency subbands at the same spatial location. Each frame is represented by a set of square wavelet blocks $\{B_{0,}\ B_{1,}\ B_2,....,B_{N_B-1}\}$. The number of wavelet blocks can be calculated by

$$N_B = \frac{W \times H}{2^L \times 2^L \times r \times r}, \qquad (1)$$

where $W \times H$ is the video frame size, $r \times r$ is the block size in the lowest band, and L is the number of DWT levels. Both W and H are assumed to be multiples of 16. Then the wavelet block size $w \times w$ can be derived directly from the block size in the lowest band and the level of DWT as

$$w = 2^L \times r. \qquad (2)$$

In this work, r is set to be 2 and L is chosen as 3 for the QCIF (176×144) format. After block mapping, SPIHT is applied to each block. Each wavelet block has different characteristics and gets a different stream length after SPIHT coding. Since most of the low frequency energy is removed from motion compensation, only a part of wavelet blocks have significant wavelet coefficients that need to be encoded.

3.2 Block Reordering

In general, most energy of an image is concentrated in the low frequency components that are located at the upper-left corner of the subband pyramid. However, the motion-compensated residuals have much more high energy components that are distributed in high frequency DWT bands than regular images. Therefore, it is inefficient to apply the zerotree wavelet coding, such as SPIHT, to the residuals directly. To increase the coding efficiency, the sub-block reordering is used before the SPIHT is applied. The block reordering procedure for processing each block B_k, $k = 0, 1, ..., N_B - 1$, is described in detail as follows.

Step 1. Initialization:

(1.1) Determine the initial significance threshold n for the whole image as

$$n = \left\lfloor \log_2(\max_{(i,j)}\{|c_{i,j}|\}) \right\rfloor \qquad \textit{for all } i, j, \qquad (3)$$

where $c_{i,j}$ is the wavelet coefficient at (i, j), $i = \{0, 1, 2, ..., W-1\}$, $j = \{0, 1, 2, ..., H-1\}$.

(1.2) Define each wavelet block B_k as insignificant before the first threshold scan.

Step 2. Significance detection: Determine each wavelet block B_k with the block size $w \times w$ as significant or not.

Case1: If B_k has been defined as significant in the previous scan, then go to step 3 refinement pass.

Case2: If $\left\lfloor \log_2 |c^k_{i,j}| \right\rfloor \geq n$, for any $i \in \{0, 1, 2, ..., w-1\}$, $j \in \{0, 1, 2, ..., w-1\}$, $c^k_{i,j} \in B_k$, then B_k is set as the significant block and go to step 4 significant pass.

Case3: Otherwise, B_k is defined as the insignificant block in this threshold scan and go to step 5 significant threshold updated.

Step 3. Refinement pass: The positions of significant sub-blocks after reordering are reserved and go to step 4.2 sub-block reordering. In other words, the reordering in later scans will not affect the already reordered results.

Step 4. Significant pass:

(4.1) Sub-block decomposition: Divide the significant block B_k into $m \times m$ sub-blocks, where m is assumed to be an exact power of 2, $B_k = \{B_{k,0}, B_{k,1}, ..., B_{k,b}, ..., B_{k,m^2-1}\}$. Each sub-block $B_{k,b}$ consists of $p \times p$ coefficients, i.e. $B_{k,b} = \{c^0_b, c^1_b, ..., c^{p^2-1}_b\}$where $p = w/m$, and c^i_b is the wavelet coefficient in sub-block $B_{k,b}$, $i = \{0, 1, 2, ..., p^2 - 1\}$. The number of significant sub-blocks in block B_k is denoted as Ns_k, and $Ns_k = 0$ in initialization.

(4.2) Sub-block reordering: For all wavelet coefficients $c^i_b \in B_{k,b}$ for any $i = \{0, 1, 2, ..., p^2 - 1\}$, $b = \{0, 1, 2, ..., m^2 - 1\}$.

(4.2.1) Obtain the norm of wavelet coefficients for each significant sub-block.

For any $i=\{0,1,2,....,p^2-1\}$, if $\lfloor \log_2 |c_b^i| \rfloor \geq n$, then the sub-block $B_{k,b}$ is significant and the norm of all significant coefficients is calculated as

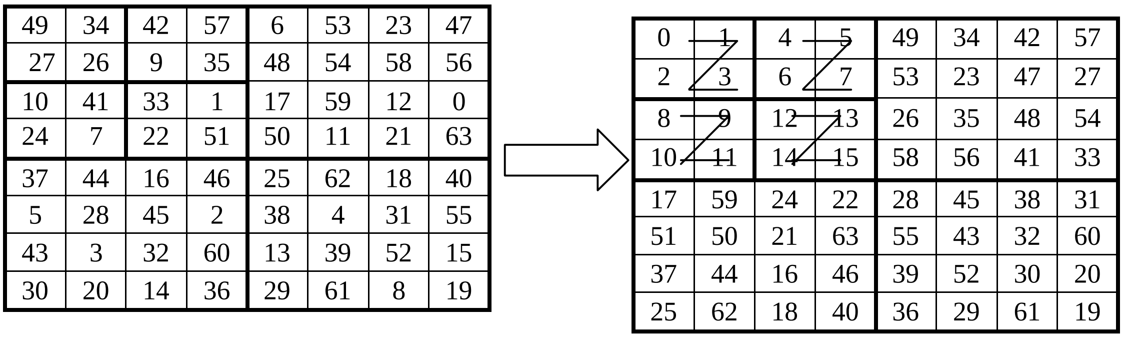
. $$Norm = \sum_{i=significant} \|c_b^i\| \quad i \in \{0,1,....,p^2-1\} \qquad (4)$$

(4.2.2) Reorder the sub-blocks based on the norm. The sub-block with the maximum norm is moved to the top left corner and other significant sub-blocks are placed following the zig-zag scan order. To reduce the overhead in recording the reordering positions, the reordering sub-block operations are restricted in the upper left quadrature of the significant sub-blocks. Let Ns_k be the counter to store the total number of significant sub-blocks including the current and previous scans, one of the following operations is performed depending on Ns_k.

- If $Ns_k \leq (\frac{1}{4} \times m^2)$, check the zig-zag scan order with the sub-block significance order. If they are the same, i.e., the sub-blocks are already located in the descent energy order, then stop the reordering in this scan and go to step 5. Otherwise, perform reordering and record the original locations of significant sub-blocks as side information, then go to step 5 significant threshold updated.
- If $Ns_k > (\frac{1}{4} \times m^2)$, check the zig-zag scan order with the sub-block significance order. If they are the same, then stop and exit the algorithm. If they are different, perform reordering and record original locations of significant sub-blocks as side information, then stop and exit the algorithm.

Step 5 Significant threshold updated: Decrease n by one and go to step 2 significance detection. If n is the least significant bit (LSB), then exit the algorithm.

Fig. 3 gives an example that describes the sub-block reordering. A significant block, size 16×16, is divided into 64 sub-blocks, $\{'0', '1', '2',....., '63'\}$, and the sub-block of number '0' represents the maximum energy sub-block while number '63' represents the minimum energy sub-block. Fig. 3(a) is the distribution of original sub-blocks, while Fig. 3(b) is the sub-block reordering result. The significant sub-blocks, $\{'0', '1', '2',....., '15'\}$, are moved to the upper left quadrature with the shift record kept, while other wavelet sub-blocks, $\{'16', '17', '18',.....,'63'\}$, are only sequentially shifted afterwards with no need to record the original positions.



49	34	42	57	6	53	23	47
27	26	9	35	48	54	58	56
10	41	33	1	17	59	12	0
24	7	22	51	50	11	21	63
37	44	16	46	25	62	18	40
5	28	45	2	38	4	31	55
43	3	32	60	13	39	52	15
30	20	14	36	29	61	8	19

0	1	4	5	49	34	42	57
2	3	6	7	53	23	47	27
8	9	12	13	26	35	48	54
10	11	14	15	58	56	41	33
17	59	24	22	28	45	38	31
51	50	21	63	55	43	32	60
37	44	16	46	39	52	30	20
25	62	18	40	36	29	61	19

(a) (b)

Fig.3: Sub-block reordering of SBR-SPIHT. A block is divided into 64 sub-blocks $\{'0', '1', '2',.....,'63'\}$ with '0' representing the maximum energy sub-block. (a) original 16×16 wavelet block. (b) reordering result.

4. Simulation Results

Simulations are performed on various video-coding algorithms, including H.263, SPIHT, as well as the proposed SBR-SPIHT methods at very low bit-rates. Class A sequences, e.g. Akiyo, Container, and Mother&Daughter, are sampled at frame rate 5 fps and encoded at bit-rate 10 kbps. Class B sequences, e.g. Foreman, News, and Silent, are sampled at frame rate 7.5 fps and encoded at bit-rate 48 kbps. All video sequences are in the QCIF (176×144) format. The H.263 encoder includes OBMC that operates at QP=10 with TMN5 test model. All simulations are running on Pentium III 800 MHz.

Table I compares intra-frame coding results for the first frames in Class A and Class B video sequences at 14 kbits per frame (0.55 bpp) with H.263, SPIHT, and SBR-SPIHT methods. For the luminance, the SBR-SPIHT outperforms the DCT-based H.263 by 2.05 dB on average. For the chrominance, the SBR-SPIHT outperforms H.263 by 1.32 dB and 1.33 dB on average for the Cb and Cr components, respectively. On the other hand, the improvement of SBR-SPIHT over SPIHT for the intra-frames is very limited because SPIHT is already able to utilize the zerotree relationship very well for still images. Table II and Table III compare inter-frame coding results for Class A and Class B video sequences at very low bit-rates with the H.263, SPIHT, and SBR-SPIHT methods. For the luminance component at 10 kbps, on average, SBR-SPIHT outperforms H.263 by 1.28 dB and SPIHT by 1.67 dB for Class A video sequences, and the processing time is about 344 ms/frame for H.263, 342 ms/frame for SPIHT, and 347 ms/frame for SBR-SPIHT, respectively. At 48 kbps, SBR-SPIHT performs better than H.263 by 0.69 dB and SPIHT by 1.3 dB on average for Class B video sequences, and the processing time is about 352 ms/frame for H.263, 351 ms/frame for SPIHT, and 360 ms/frame for SBR-SPIHT, respectively. The simulation results show that only limited time is needed for the block reordering in SBR-SPIHT. Similar improvements are obtained for the chrominance of both the Class A and Class B video sequences. Fig. 4 compares the PSNRs of the luminance component of each frame of SPIHT, H.263, and SBR-SPIHT with Mother&Daughter video sequence. They are compared under the circumstance that all three methods are operated at the same average bit-rate.

Fig. 5 and Fig. 6 show the luminance component of reconstructed frame by using H.263 and SBR-SPIHT in Class A and Class B video sequences, respectively. The blocking artifacts are generated by H.263 at very low bit-rates are shown as in Fig. 5(a). The visual quality of SBR-SPIHT is clearly superior in the edges of reconstructed frames such as Mother's face and shoulder in Fig. 5(b) and Reporter's face in Fig. 6(b). Since SBR-SPIHT performs wavelet transform on the whole image, the blocking artifact is not serious. On the other hand, the DCT based image coding methods, such as H.263, suffers more serious artifact. These results reveal that the proposed algorithm offers a more efficient way to encode the significant wavelet coefficients than the original SPIHT. The SBR-SPIHT can be efficiently applied to video residuals by concentrating the encoding of randomly dispersed energy in local areas.

5. Conclusions

We have presented SBR-SPIHT coding that uses a novel selective block-wise reordering technique for video coding. The approach is particularly suitable for coding video sequences at very low bit-rates. The proposed method provides the advantage that the energy compaction in the upper left corner indeed improves the SPIHT coding efficiency by reducing the sorting cost and increasing the bit budget for actual reconstruction. The proposed block reordering algorithm with the wavelet transform has shown a solution for the long time problem that the zerotree based wavelet coding does not perform as well for videos as for images.

REFERENCES

[1] Streaming video Profile-Final Draft Amendment (FDAM4), MPEG01/N3904.

[2] Draft ITU-T Recommendation H.263, Video coding for low bit rate communication, May 1996.

[3] S. A. Martucci, I. Sodagar, T. Chiang, and Y. Q. Zhang, "A zerotree wavelet video coder," IEEE Trans. on Circuits and Syst. Video Technol., vol. 7, pp. 109-118, Feb. 1997.

[4] J. Vass, B. B. Chai, and X. Zhuang, "Significance-linked connected component analysis for very lower bit-rate wavelet video coding," IEEE Trans. on Circuits and Syst. Video Technol., vol. 9, pp. 630-647, June 1999.

[5] K. Shen and E. J. Delp, "Wavelet based rate scalable video compression," IEEE Trans. on Circuits and Syst. Video Technol., vol. 9, pp. 109-122, Feb. 1999.

[6] G. Xing, J. Li, S. Li, and Y. Q. Zhang, "Arbitrarily shaped video-object coding by wavelet," IEEE Trans. on

Circuits and Syst. Video Technol., vol. 11, pp. 1135-1139, Oct. 2001.

[7] 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 Syst. Video Technol., vol. 6, pp.243-250, June 1996.

[8] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” IEEE Trans. Signal Processing, vol. 41, pp. 3445-3462, Dec. 1993.

[9] T. T. Lu, K. W. Wen, and P. C. Chang, “Block reordering wavelet packet SPIHT image coding,” in Proc. IEEE Pacific-Rim Conference on Multimedia, pp. 442-449, Oct. 2001.

[10] B. J. Kim, Z. Xiong and W. A. Pearlman, “Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees,” IEEE Trans. on Circuits and Syst. Video Technol., vol. 10, pp. 1374-1387, Dec. 2000.

[11] K. K. Lin and R. M. Gray, “Video residual coding using SPIHT and dependent optimization,” in Proc. Data Compression Conf. pp. 113-122, 2001.

[12] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” Commun. ACM, vol. 30, pp. 520-540, June 1987.

[13] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, “Image coding using wavelet transform,” IEEE Trans. on Image Processing, vol. 1, pp. 205-220, April 1992.

Table I: Performance comparison of intra-frame coding at 14 kbits per frame

Sequence	Bits	Luminance, PSNR [dB]			Cb, PSNR [dB]			Cr, PSNR [dB]		
		H.263	SPIHT	SBR-SPIHT	H.263	SPIHT	SBR-SPIHT	H.263	SPIHT	SBR-SPIHT
Akiyo	14000	33.06	35.32	35.45	35.37	36.55	36.67	37.25	38.33	38.56
Container	14088	28.81	30.23	30.48	36.40	37.43	37.61	35.02	36.10	36.32
Mother& Daughter	13624	33.78	36.22	36.45	39.12	40.52	40.72	39.54	41.12	41.44
Foreman	13976	30.11	32.12	32.31	38.14	38.66	38.94	38.23	38.95	39.23
News	14080	28.60	29.93	30.12	33.16	34.68	34.87	34.48	35.45	35.67
Silent	13688	30.34	31.98	32.16	34.63	35.72	35.93	36.40	37.53	37.65

Table II: Performance comparison of inter-frame coding for Class-A sequences at 5 fps and 10 kbps

Sequence	Bit-rate (kbps)	Luminance, PSNR [dB]			Cb, PSNR [dB]			Cr, PSNR [dB]		
		H.263	SPIHT	SBR-SPIHT	H.263	SPIHT	SBR-SPIHT	H.263	SPIHT	SBR-SPIHT
Akiyo	8.61	35.02	34.75	36.56	37.59	38.15	38.95	39.81	41.44	42.21
Container	7.92	30.72	30.28	32.09	37.58	38.34	39.25	36.72	37.52	38.34
Mother& Daughter	7.95	33.31	32.85	34.24	39.42	39.03	40.63	40.08	40.06	41.06

Table III: Performance comparison of inter-frame coding for Class-B sequences at 7.5 fps and 48 kbps

Sequence	Bit-rate (kbps)	Luminance, PSNR [dB]			Cb, PSNR [dB]			Cr, PSNR [dB]		
		H.263	SPIHT	SBR-SPIHT	H.263	SPIHT	SBR-SPIHT	H.263	SPIHT	SBR-SPIHT
Foreman	46.48	31.57	30.97	32.12	37.31	36.37	37.57	37.40	36.23	37.82
News	45.78	34.84	34.11	35.57	38.61	37.24	38.87	39.21	38.67	39.81
Silent	46.15	35.17	34.67	35.96	38.38	38.72	39.14	39.54	39.43	40.63

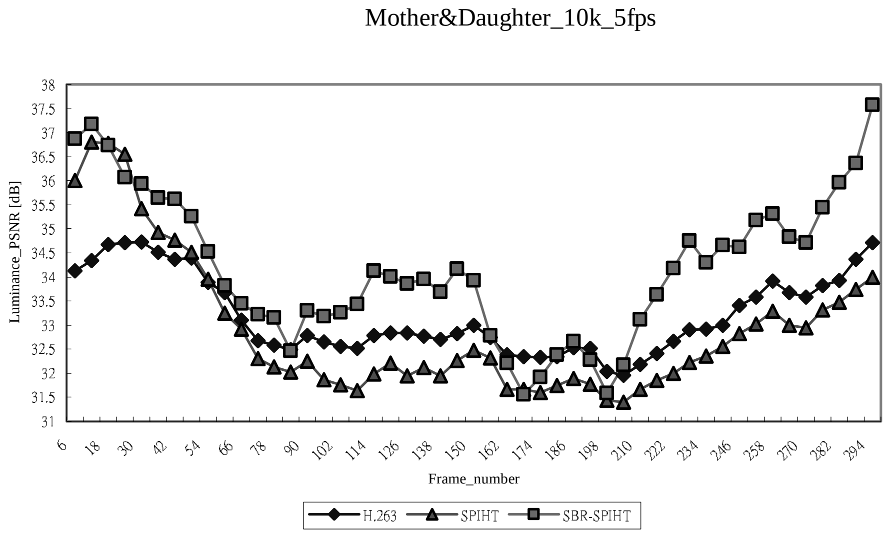
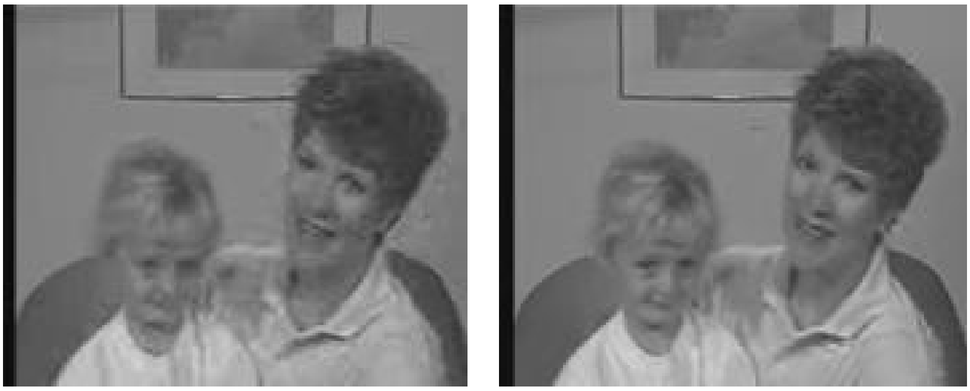
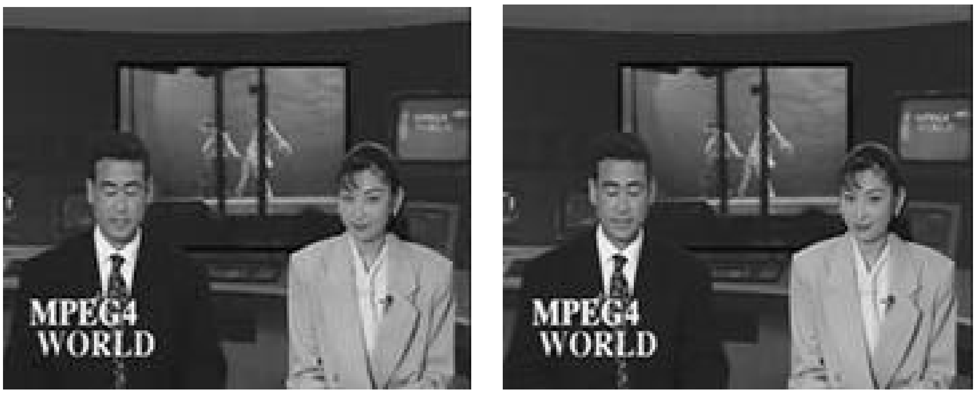


Fig.4: PSNR performance comparisons of H.263, SPIHT, and SBR-SPIHT for Class A video sequence at 5 fps and 10 kbps.



(a) (b)

Fig.5: The 252th reconstructed frame of Mother&Daughter (QCIF format) by (a) H.263 (b) SBR-SPIHT at 5 fps and 10 kbps.



(a) (b)

Fig.6: The 116th reconstructed frame of News (QCIF format) by (a) H.263 (b) SBR-SPIHT at 7.5 fps and 48 kbps.