Fast and Efficient 3-D Wavelet Based Video Coding Technique

Athar Ali Moinuddin, Ekram Khan and Mohammed Ghanbari

Abstract— In this paper, an efficient and embedded 3-D wavelet based video coding technique is proposed. The key idea is the use of a composite block-tree hierarchical structure to link blocks of wavelet coefficients in spatial, temporal and color planes in such a way that a large number of the insignificant sets are coded together. The proposed scheme uses $n \times n$ block of wavelet coefficients as a basic unit, in contrast to the existing 3-D set partitioning in hierarchical trees (3-D SPIHT) algorithm that uses a single coefficient as a basic unit. Simulation results show a significant performance improvement in coding efficiency and reduction in computational complexity over the 3-D SPIHT video coder.

Index Terms— **Video coding, 3-D wavelet video coding, 3-D tree structure, color coding.**

I. Introduction

Video delivery through heterogeneous networks over new multimedia devices of varying capabilities requires scalable coding. In recent years, scalable video coding using the 3-D wavelet transform has gain a lot of attention [1]. Wavelet based 3-D video coding systems use spatio-temporal analysis of a group of frames (GOF) followed by coefficients encoding. An efficient spatio-temporal analysis and coefficient encoding is required to achieve effective compression performances.

Inspired by the success of embedded zerotree wavelet (EZW) [2] and set partitioning in hierarchal trees (SPIHT) [3] algorithms for compression of gray scale images, they are successfully extended for 3-D video coding [4]-[6]. The successful extension of SPIHT for 3-D video coding is proposed in [5]. An efficient tree structure for packet decomposition using 3-D SPIHT is developed in [6]. Moreover, in these codecs, the color video sequences are coded by treating luminance-chrominance (YUV) color planes independently, assuming that they are mutually exclusive. In the scheme proposed in [5], during the significance testing in each bit-plane, first all the Y plane coefficients are checked followed by all the U plane coefficients and then all the V plane coefficients. However, this approach is more biased towards the luminance component. The problem is that if the bit-budget is exhausted at the beginning of a particular bit-plane, some of the bits might be wasted in coding the insignificant luminance coefficients which could otherwise be utilized to code many significant chrominance coefficients.

In color video sequences, the three-color planes may be uncorrelated, but they are not independent. A color video coding algorithm using the EZW framework and exploiting the interdependency among the three-color planes was first proposed in [7]. In this algorithm, each luminance node is having up to six children, four in the luminance plane and one in each of the two chrominance planes. However, it suffers with the problem of early accumulation of chrominance nodes in the list, thereby reducing its coding efficiency. Many efficient and embedded color coding techniques based on SPIHT are proposed [8]-[10], in which the three color planes are linked through a tree structure with root nodes in luminance (Y) color plane only. In these algorithms, a root node has all of its descendents either in the luminance or chrominance planes only.

In this paper, we propose an efficient and embedded 3-D wavelet video coding technique. It is based on partitioning a spatio-temporally transformed video frames into coefficient blocks, each of size $n \times n$. A comprehensive spatial-temporal orientation tree of the blocks (known as block-tree) is then used to exploit the self-similarity and magnitude localization property in the spatio-temporal subbands as well as in the luminance-chrominance color planes. In a block-tree, significant blocks are found using the tree partitioning concept of SPIHT [3], whereas significant coefficients within each significant block are found using the quad-tree partitioning of SPECK [11]. A significant block-tree is recursively partitioned (with combined tree and block partitioning) until the significant coefficients are found.

The rest of the paper is organized as follows. The proposed coding technique is described in section II. Simulation results and discussions are presented in section III and finally the paper is concluded in section IV.

II. The Proposed Technique

The basic structure of the 3-D video coder is shown in Fig. 1 which essentially consists of: spatio-temporal analysis using 3-D wavelet transform, 3-D tree structure for the wavelet coefficients and coefficients encoding algorithm. In the spatio-temporal analysis, temporal decomposition followed by spatial decomposition is applied on each GOF. In the proposed technique, each of the three color planes

Manuscript received March 6, 2008. Athar Ali Moinuddin wishes to acknowledge the financial support provided by the Association of Commonwealth Universities, UK and Aligarh Muslim University, Aligarh INDIA.

Athar Ali Moinuddin is with the Department of Computing and Electronic Systems, University of Essex, Colchester, CO4 3SQ, UK (phone: +44-7954967792; e-mail: aamoin@ essex.ac.uk).

Ekram Khan is with the Department of Electronics Engineering, Aligarh Muslim University, Aligarh, 202002, INDIA (e-mail: ekhan67@gmail. com).

Mohammed Ghanbari is with the Department of Computing and Electronic Systems, University of Essex, Colchester, CO4 3SQ, UK (e-mail: ghan@essex.ac.uk).

(4:2:0 YUV format) is wavelet transformed using N_s levels of decomposition for the luminance (Y) plane and *(N_s -1)* levels for each of the chrominance (U, V) planes. Since in 4:2:0 color format, the resolutions of the chrominance planes are one-quarter to that of the luminance plane, wavelet decomposition of the chrominance planes by one level less than that of the luminance plane will result in the LL-band of each transformed planes of the same dimensions. This will simplify the child-parent relationship to link the three-color planes together.

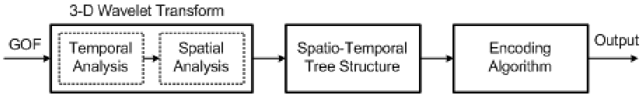

Fig. 1. Basic structure of 3-D wavelet video coder

After spatio-temporal analysis of a GOF, the wavelet coefficients in each color plane are divided into blocks of $n \times n$ coefficients. To exploit the self-similarity and magnitude localization property in the spatio-temporal subbands, a spatial orientation tree of blocks (block-tree) is used. In a significant block-tree, significant blocks are found using the tree partitioning concept of SPIHT, whereas significant coefficients within each significant block are found using quad-tree partitioning of SPECK. A significant block-tree is recursively partitioned until significant coefficients are found. Additionally, to exploit the interdependency of the color planes, the block-trees of the three color planes are linked together through a composite spatial orientation tree of blocks as shown in Fig. 2. In order to maintain the clarity, only four temporal frames with two levels of temporal decompositions are shown. However, this basic structure can be extended to N_t levels of temporal decomposition on a GOF size of 2^{N_t} . Also shown in Fig. 2 is the associated U and V color planes with each of Y plane. The parent child relationship can be described as follows. Let

$B(i, j, t : X)$: an arbitrary block of size $n \times n$ at column *i*, row *j* and frame *t* in color plane *X*, $X \in \{Y, U, V\}$
w: width of the lowest frequency spatial subbands
h: height of the lowest frequency spatial subbands
f: width of the lowest frequency temporal subband
$O(i, j, t : X)$ = offsprings of a block $B(i, j, t : X)$

- if $X = Y$, then
  - ∗ if $i < w$, $j < h$, $t < f$

$$O(i,j,t:Y)=\left\{\begin{array}{l}B(i+w,j,t:Y),B(i,j+h,t:Y),B(i+w,j+h,t:Y),\\ B(i,j,t:U),B(i,j,t:V),B(i,j,t+f:Y)\end{array}\right\}$$

  - ∗ else if $i < w$, $j < h$, t in the highest frequency temporal subband

$$O(i,j,t:Y)=\left\{\begin{array}{l}B(i+w,j,t:Y),B(i,j+h,t:Y),B(i+w,j+h,t:Y),\\ B(i,j,t:U),B(i,j,t:V)\end{array}\right\}$$

  - ∗ else if $i < w$, $j < h$

$$O(i,j,t:Y)=\left\{\begin{array}{l}B(i+w,j,t:Y),B(i,j+h,t:Y),\\ B(i+w,j+h,t:Y),B(i,j,t:U),\\ B(i,j,t:V),B(i,j,2t:Y),B(i,j,2t+1:Y)\end{array}\right\}$$

  - ∗ else

$$O(i,j,t:Y)=\left\{\begin{array}{l}B(2i,2j,t:Y),B(2i+1,2j,t:Y),\\ B(2i,2j+1,t:Y),B(2i+1,2j+1,t:Y)\end{array}\right\}$$

- else if $X \in \{U, V\}$
  - ∗ if $i < w$, $j < h$

$$O(i,j,t:X)=\left\{\begin{array}{l}B(i+w,j,t:X),B(i,j+h,t:X),\\ B(i+w,j+h,t:X)\end{array}\right\}$$

  - ∗ else

$$O(i,j,t:X)=\left\{\begin{array}{l}B(2i,2j,t:X),B(2i+1,2j,t:X),\\ B(2i,2j+1,t:X),B(2i+1,2j+1,t:X)\end{array}\right\}$$

We have used a bit-plane based coding algorithm comprising of two main stages within each bit-plane; sorting and refinement passes. The coding process starts with the most significant bit plane and proceeds towards the finest resolution. The coefficients significance is managed by the use of three ordered lists: a list of insignificant blocks (LIB), a list of insignificant block sets (LIBS) and a list of significant pixels (LSP). At the initialization step, only the root blocks from the lowest temporal band of Y plane are added to LIB and LIBS. The LSP starts as an empty list.

During the sorting pass, the encoder first traverses through LIB, testing the significance of a block against the current threshold. For each block in the LIB, one bit is used to describe its significance. If the block is not significant, then it is a zero-block and a '0' is sent, it remains in the LIB and no more bits will be generated. Here, insignificant information of $n \times n$ individual coefficients is conveyed using a single '0' bit, whereas in 3-D SPIHT this will generate $n \times n$ '0' bits. Otherwise, if the block is significant then a '1' is sent and it is partitioned into four adjacent blocks (quad-tree partitioning). The partitioning operation is repeated recursively until no further partition is needed or the smallest possible block size (individual coefficient) is attained. At this stage four coefficients and their significances are tested individually. If a coefficient is insignificant, then a '0' is sent and it is moved to LIB as a single coefficient block. Otherwise, if a coefficient is significant, then a '1' is sent and its sign bit is also coded and the coefficient is moved to LSP. After testing all the four individual coefficients in the block, the current block is deleted from LIB.

Similarly, each set in LIBS requires one bit for significance information. Insignificant sets remain in LIBS while the significant set will be partitioned into subsets. A significant type 'A' set will be partitioned into a type 'B' set and its offspring blocks. The type 'B' set is added to the end of LIBS while the offspring blocks are immediately examined for significance as is done in LIB. A significant type 'B' set will be partitioned into type 'A' sets; all of them are added to the end of LIBS.

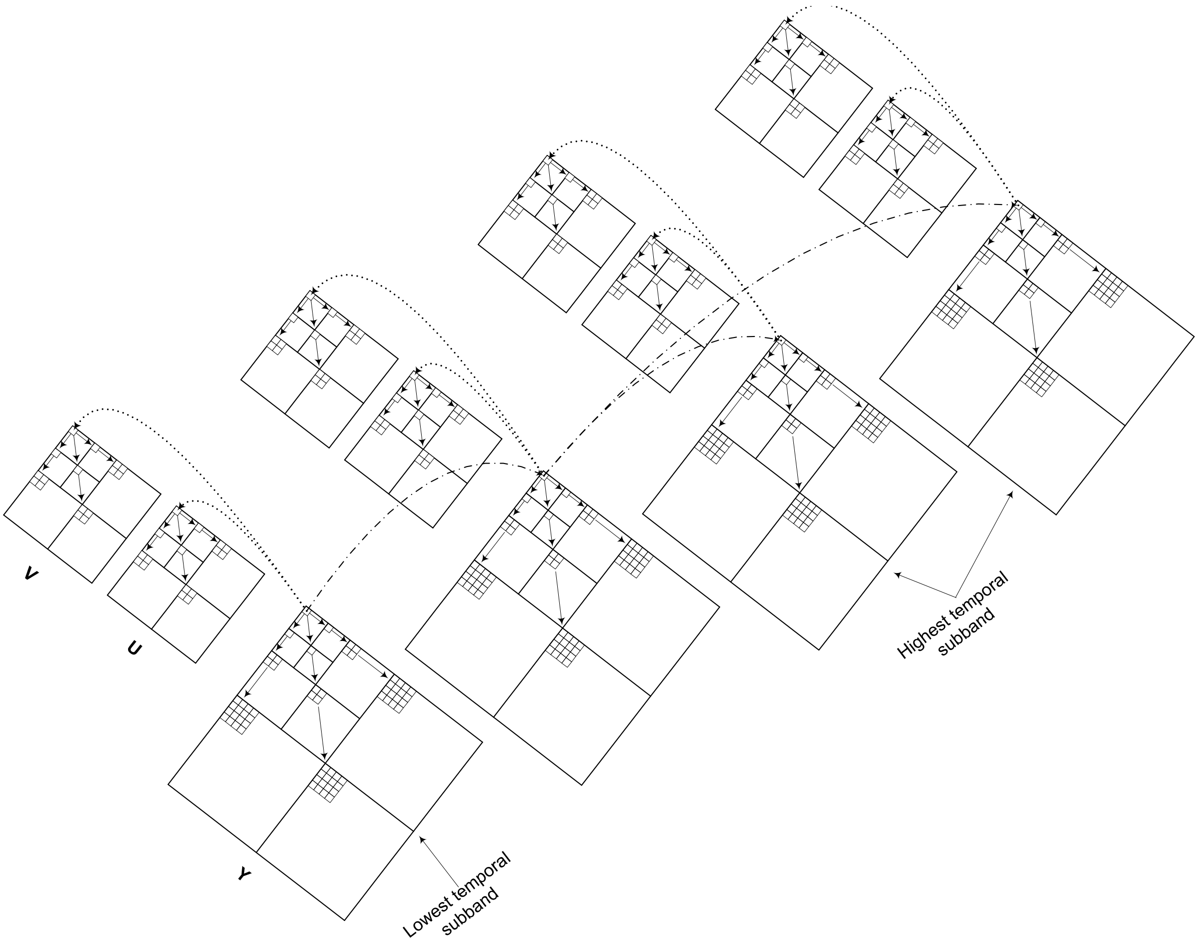

Fig. 2. Parent-Child relationship in the block-tree structure

Since all of the newly generated insignificant sets are added to the end of LIBS, they will be processed in the same manner at the same threshold until each one of them is examined. After each sorting pass, each coefficient in LSP, except those just added in the current bit plane, is refined with one bit. The algorithm then repeats the above procedure by decreasing the current threshold by a factor of two until the desired bit rate is achieved.

III. Simulation Results

Performance of the proposed technique is evaluated on CIF (352×288) test sequences *Salesman* and *Akiyo*, each of 96 frames. For 3-D wavelet transform, a GOF of 16 frames is first temporally decomposed into four temporal subbands using 5/3-biorthogonal filters. The resulting frames are spatially decomposed using 9/7-biorthogonal filters [12] as follows. For Y plane, we used four levels of decomposition, and each of U and V planes is decomposed with three levels only. All the tests were performed at the frame rate of 30 frames per second. The results are compared with the 3-D SPIHT algorithm, in which each color plane is spatially decomposed to 4 levels.

Our implementation of 3-D SPIHT is similar to the one given in [5] but with the following differences. The CIF resolution images with four levels of decompositions result in odd dimensions of LL-band of chrominance planes and hence it is not suitable for SPIHT-type child-parent relationship. Kim *et al*. have solved this problem by extending the chrominance planes before the 2-D spatial transformation [5]. However, we believe that this increases the complexity of the encoder and decoder and slightly reduces the coding efficiency also. In our implementation, the last rows and columns of the LL-band of chrominance planes follow the EZW's tree structure. Our 3-D SPIHT also uses separate coding of each color plane as in [5], but initialization structure of LIP and LIS is as done in [8]. The initial block size in the proposed technique is considered as 2×2. All results are without arithmetic coding and without motion compensation in both coders. The objective quality of the decoded frames in terms of the peak-signal-to-noise-ratio *(PSNR)* in *dB* of the three color planes *X*, $X \in \{Y, U, V\}$ is defined as

$$PSNR_X = 10\log_{10}\frac{255^2}{mse(X)} \quad (1)$$

The implementations were done in *C* programming language under LINUX operating system. Tests were performed using a personal computer with AMD Athlon 64 processor with CPU speed of 2.21 GHz and 2 GB RAM.

Figures 3 and 4 show the average luminance PSNR (*dB*) for the *Salesman* and *Akiyo* sequences. A comparison reveals that the proposed technique always outperforms 3-D SPIHT. In particular, it brings a performance gain of about 0.4-2.1 *dB* with respect to 3-D SPIHT for both sequences. In Table 1, we compare the average PSNR (*dB*) of the three color planes at two different bit budgets. From the analysis of these results it is clear that improvement in PSNR of Y plane in the proposed coding technique is not at the expense of the chrominance planes. The PSNR results of the U and V planes are comparable in both codecs. This is in spite of using one level less spatial decomposition for the chrominance planes in the proposed techniques as compared to 3-D SPIHT. Since human eye is more sensitive to the changes in the brightness, therefore the gain for Y plane as obtained from the proposed coder will yield visually better results. The reason for the superior performance is due to the better exploitation of intra-band correlations in the form of zeroblock as well as the interdependency of the color planes.

We also compare the computational complexity of the two coders by measuring the run times of encoding and decoding. Table 2 summarizes these timings for the two codecs. It can be observed that the proposed algorithm encodes a set of wavelet coefficients about 55-64 % faster than 3-D SPIHT. On the other hand its decoder timing is comparable to that of 3-D SPIHT. The possible reason for this is that being a block-based encoder it has considerably smaller number of elements in its lists than in 3-D SPIHT. Therefore, it has considerably reduced memory access timings which make it faster than 3-D SPIHT.

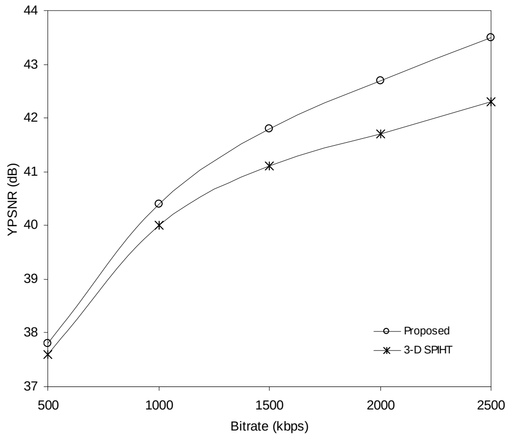

Fig. 3. Rate-distortion comparison at various bit rates for test sequence *Salesman.*

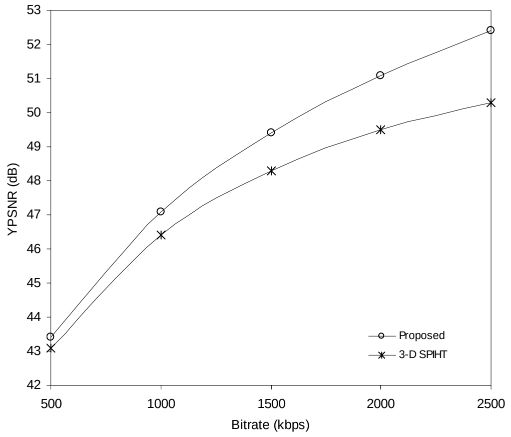

Fig. 4. Rate-distortion comparison at various bit rates for test sequence *Akiyo*.

IV. Conclusions

In this paper, an embedded 3-D wavelet video coding technique is proposed. It is based on the joint exploitation of inter- and intra-subband correlations of the wavelet coefficients within each color plane and inter-dependency among the luminance-chrominance color planes. The proposed technique has been shown to provide a better coding efficiency and has low computational complexity compared to 3-DSPIHT. Use of motion compensation and arithmetic coding can provide further enhancement in the performance. Also, the bit streams can easily be tailored for temporal and spatial scalability requirements.

TABLE 1
Rate-Distortion Comparison (color plane wise)

Bitrate (kbps)	3-D SPIHT			Proposed		
	Y	U	V	Y	U	V
Salesman						
1000	40.0	46.5	**47.8**	**40.4**	**46.6**	**47.8**
2000	41.7	48.2	49.5	**42.7**	**48.3**	**49.6**
Akiyo						
1000	46.4	**51.5**	52.3	**47.1**	**51.5**	**52.4**
2000	49.5	**55.1**	**55.7**	**51.1**	**55.1**	**55.7**

TABLE 2
Comparison of Encoding and Decoding Timings

Bitrate (kbps)	Encoding time (ms)		Decoding time (ms)	
	3-D SPIHT	Proposed	3-D SPIHT	Proposed
Salesman				
1000	2190	970	80	80
2000	2800	1010	140	150
Akiyo				
1000	2410	940	70	70
2000	3000	1350	130	140

References

[1] N. Adami, A. Signoroni, and R. Leonardi, “State-of-the-art and trends in scalable video compression with wavelet-based approaches,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, pp. 1238-1255, Sept. 2007.

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

[3] A. Said and W. A. Pearlman, “A new, fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.

[4] P. Campisi, M. Gentile, and A. Neri, “Three dimensional wavelet based approach for a scalable video conference system,” *IEEE International Conference on Image Processing (ICIP’99)*, vol. 3, pp. 802–806, 1999.

[5] 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. Circuits and Systems for Video Technology, vol. 10, pp. 1374-1387, Dec. 2000.

[6] C. He, J. Dong, Y. F. Zheng, and Z. Gao, “Optimal 3-D coefficient tree structure for 3-D wavelet video coding,” IEEE Trans. Circuits and Systems for Video Technology, vol. 13, pp. 961-972, Oct. 2003.

[7] K. Shen and E. J. Delp, “Wavelet based rate scalable video compression,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 9, pp. 109-122, February 1999.

[8] E. Khan and M. Ghanbari, “Efficient SPIHT-based embedded color image coding techniques,” *IEE Electronics letters*, vol. 37, pp. 951-952, July 2001.

[9] A. A. Kassim, and W. S. Lee, “Embedded color image coding using SPIHT with partially linked spatial orientation trees,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp. 203-206, Feb. 2003.

[10] A. A. Kassim, W. S. Lee, “Performance of the color set partitioning in hierarchical tree scheme in video coding,” Circuits Systems Signal Processing, vol. 20, pp. 253-270, March 2001.

[11] W. A. Pearlman, A. Islam, N. Nagaraj and A. Said, “Efficient low-complexity image coding with Set-Partitioning Embedded Block Coder,” *IEEE Trans. Circuits Sys. Video Tech.*, Vol. 14, pp. 1219-1235, Nov. 2004.

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