IMAGE AND VIDEO CODING WITH A NEW WASH TREE ALGORITHM FOR MULTIMEDIA SERVICES

[1]G.SURESH, [2]P.EPSIBA, [3]DR.M.RAJARAM, [4]DR.S.N.SIVANANDAM

[1]Research Scholar, CAHCET, Vellore, India.
[2]Lecturer, Department of ECE, CAHCET, Vellore, India.
[3]Professor & Head, Department of EEE, GCE, Thirunelveli, India.
[4]Professor & Head, Department of CSE, PSG Tech, Coimbatore, India.

ABSTRACT

In this paper, a weighted adaptive scalable hierarchical (WASH) tree based video coding algorithm is proposed with low-memory usage. The standard coding uses three separate lists to store and organize tree data structures and their significance, which can grow large at high rate and consume large amounts of memory. In the proposed algorithm, value added adaptive scale down operator discards unnecessary lists and the process length of the sorting phase is shortened to reduce coding time and memory usage. Spatial and temporal scalability can be easily incorporated into the system to meet various types of display parameter requirements and self-adapting rate allocations are automatically achieved. Results show that the proposed method reduces memory usage, run time and improves PSNR.

Keywords: *Low Memory Set Partition In Hierarchical Tree, Modified SPIHT, Scalable Video Coding*

1. INTRODUCTION

Network bitrates continue to increase (dramatically in the local area and somewhat less so in the wider area), high bitrate connections to the home are commonplace and the storage capacity of hard disks, flash memories and optical media is greater than ever before. With the price per transmitted or stored bit continually falling, it is perhaps not immediately obvious why video compression is necessary (and why there is such a significant effort to make it better). Video compression has two important benefits. First, it makes it possible to use digital video in transmission and storage environments that would not support uncompressed ('raw') video. Second, video compression enables more efficient use of transmission and storage resources. If a high bitrate transmission channel is available, then it is a more attractive proposition to send high-resolution compressed video or multiple compressed video channels than to send a single, low-resolution, uncompressed stream. Even with constant advances in storage and transmission capacity, compression is to be a very essential component of multimedia services. Image and video compression has been a very active field of research and development for over 20 years and many different systems and algorithms for compression and decompression have been proposed and developed. In order to encourage interworking, competition and increased choice, it has been necessary to define standard methods of compression encoding and decoding to allow products from different manufacturers to communicate effectively. This has led to the development of a number of key International Standards for image and video compression, including the JPEG, MPEG and H.26× series of standards [11]. Each standard is a document that primarily defines two things, a coded representation (or syntax) that describes visual data in a compressed form and a method of decoding the syntax to reconstruct visual information. Among image and video compression methods, the SPIHT [1] algorithm has been broadly used and has became an interesting alternative to the current H.264 AVC standard. The SPIHT algorithm applies wavelet transform to image and video, organizes the coefficients in tree data structures, and partitions the wavelet coefficients according to their significance in the trees. However, the main drawback of the SPIHT algorithm is that it requires a large amount of time and memory to accurately accomplish the job. As a result, it can cause a delay or pause in real time video transmission, which requires large amounts of bandwidth to begin with. Thus, it is important to optimize the algorithm to provide a better performance for real time applications. There are two popular wavelet algorithms for 3D compression: 3D-IEZW and 3D SPIHT. The embedded zero-tree wavelet (EZW) algorithm, which was introduced by Shapiro [2], is a coding system that searches through subbands and transmits the transformed coefficients as a zerotrees. On the other hand, B. J Kim et. al. extended the 2D SPIHT into three-dimensions (3D-SPIHT) in [3-4]. It is important to note that, as a byproduct of a wavelet transform, wavelet coefficients in different bands can be used to predict motion [2]. This is in contrast to the H.264 AVC [12] standard where costly motion estimation and compensation algorithms are needed.

In this paper 3D tree algorithm structure is optimized to reduce total encoding time. This paper displays exceptional characteristics like,
• Good image quality with a high peak signal-to-noise ratio (PSNR); • Fast coding and decoding;
• A fully progressive bit-stream; • can be used for lossless compression; • may be combined with error protection; • Ability to code for exact bit rate or PSNR.

2. PREVIOUS WORK

A. The SPIHT Coding Algorithm

The 2-D SPIHT algorithm [5], like the embedded zerotree wavelet (EZW) coding algorithm [2], views wavelet coefficients as a collection of spatial orientation trees, with each tree consisting of coefficients from all subbands that correspond to the same spatial location in an image. It uses multipass “zerotree” coding to transmit the largest wavelet coefficients (in magnitude) first. A set of tree coefficients is significant if the largest coefficient magnitude in the set is greater than or equal to a certain threshold (e.g., a power of two); otherwise, it is insignificant. Similarly, a coefficient is significant if its magnitude is greater than or equal to the threshold; otherwise, it is insignificant. In each pass, the significance of a larger set in the tree is tested first: if the set is insignificant, a binary “zerotree” bit is used to set all coefficients in the set to zero; otherwise, the set is partitioned into subsets (or child sets) for further significance tests. After all coefficients are tested in one pass, the threshold is halved before the next pass. The underlying assumption of SPIHT coding is that most images can be modeled as having decaying power spectral densities. That is, if a parent node in the wavelet coefficient tree is insignificant, it is very likely that its descendents are also insignificant. The zerotree symbol is used very efficiently in this case to signify a spatial subtree of zeros. When the thresholds are powers of two, SPIHT [7-8] coding can be thought of as a bit-plane coding scheme. It encodes one bit-plane at a time, starting from the most significant bit. With the sign bits and refinement bits (for coefficients that become significant earlier) being coded on the fly, SPIHT achieves *embedded* coding in the wavelet domain using three lists: 1) the list of significant pixels (LSP); 2) the list of insignificant pixels (LIP); and 3) the list of insignificant sets (LIS). The 2-D SPIHT coder performs competitively with most other coders published in the literature [3], while possessing desirable features such as relatively low complexity and rate embeddedness. It represents the current state-of-the-art of wavelet image coding.

B. 3-D SPIHT

The 2-D SPIHT algorithm [1] was extended to 3-D embedded SPIHT video coding in [9] and [10]. Besides motion compensation, the 3-D SPIHT algorithm is, in principle, the same as 2-D SPIHT, except that 3-D wavelet coefficients are treated as a collection of 3-D spatio-temporal orientation trees [see Fig.1] and that context modeling in arithmetic coding is more involved. A block-based motion estimation scheme is implemented in the 3-D SPIHTcoder in [9] and [3] and an option for not using motion estimation is also allowed to reduce theencoding complexity. Global affine motion compensation was combined with the 3-D SPIHT algorithm in [10]. Every 16 frames form a group of pictures for 3-D wavelet transformation. Extension to color is accomplished without explicit bit allocation and can be used for any color space representation. Spatio-temporal orientation trees coupled with powerful SPIHT sorting and refinement turns out to be very efficient. In addition to being rate scalable, the 3-D SPIHT video coder allows multiresolutional scalability in encoding and decoding in both time and space[14]. This added functionality, along with many desirable features, such as full embeddedness for progressive transmission, precise rate control for constant bit-rate traffic, and low complexity for possible software-only video applications, makes the new video coder an attractive candidate for multimedia applications like Internet video.

C. LM SPIHT

In this method the enhanced 2D SPIHT introduced by Y. Sun et. al. [5] and extended into three-dimensions. Given that the SPIHT coding uses three separate lists to store and manage trees data structures and their significance, these lists can rapidly grow large and consume high amount of memory. As presented in [5], they used only one list to organize tree data structures and output the first (*n*+1-error bits) bits immediately after the significant test. Here the number of error bits and absolute zerotree and converted them into 3D to trim down unnecessary parts of the 3D SPIHT algorithm. The error bits indicate the number of bits that are thrown away and the bits, including all its descendants that are just truncated, become absolute zerotrees which will never be significant.

3. PROPOSED METHOD

In the proposed method, we used the adaptive scalable hierarchical tree algorithm for video coding. This approach is quite attractive since we do not need to change the encoder structure. We achieve full scalability in rate and partial scalability in space and time with multiresolution encoding and decoding. Here the major modification is concentrated over the scalability property, both in precoding and postcoding levels. Obvious advantages of adaptive scalable video coding are savings in memory and decoding time. In a practical implementation two simple scalable operators (equation (1),(2)) were derived, which made an effective changes with every frame based on its density.

f^(i,j)=f(i,j)-0.25*(f(i+t,j+t)-f(i-t,j-t)-f(i-t,j+t)
-f(i+t,j-t)) (1)
f^(i,j)=f(i,j)-0.25*(f(i+t,j)-f(i,j+t)-f(i,j-t)-f(i-t,j)) (2)

Implementation steps:

1. Initialization 2. prescaling operators
3. sorting pass 4. refinement pass
5.quantization step update

Algorithm:

Step 1: (Initialization)
Output set the LSP as an empty list, add the coordinates to the LIP, add the coordinates with descendants to the list LIS, as type A entries,
Step 2: Pass through scalable operators
Step 3: (Sorting Pass)
3.1) for each entry (i, j) in the LIP do: output Sn(i, j), if Sn(i, j)=1
move (i, j) to the LSP, output the sign of $c_{i,j}$,
3.2) for each entry (i, j) in the LIS do:
3.2.1) if the entry is of type A then output Sn(D(i, j)), if Sn(D(i, j)) = 1 then
* for each $(k, l) \in O(i, j)$ do: output Sn(k, l),
if Sn(k, l) = 1 then add (k, l) to the LSP, output the sign of $c_{k,l}$,
if Sn(k, l)=0 then add (k, l) to the end of the LIP,
*if $L(i, j) \neq 0$ then move (i, j) to the end of the LIS, as an entry of type B,
go to Step 3.2.2).
otherwise
remove entry (i, j) from the LIS,
3.2.2) if the entry is of type B output Sn (L(i, j)),
if Sn(L(i, j)) = 1 then *add each $(k, l) \in O(i, j)$ to the end of the LIS as an entry of type A,
*remove (i,j) from the LIS,
Step 4: (Refinement Pass)
For each entry (i, j) in the LSP, except those included in the last sorting pass
(i.e., with the same n),
output the *n*th most significant bit of $|c_{i,j}|$,
Step 5: (Quantization-Step Update) Decrement *n* by 1 and go to Step 2.

4. EXPERIMENTAL RESULTS

To compare the performance of the proposed technique, 3D SPIHT, 3D SPIHT Low-Memory(LM) and proposed adaptive scalable hierarchical tree algorithms were implemented in MATLAB. A graphical user interface (GUI) was built to make it easier to monitor the impact of changing parameters, such as frame size, compression ratio and PSNR as the program is run. Five different video files were employed to examine and analyze the proposed model. The experimental results and comparisons between all these three algorithms are displayed.

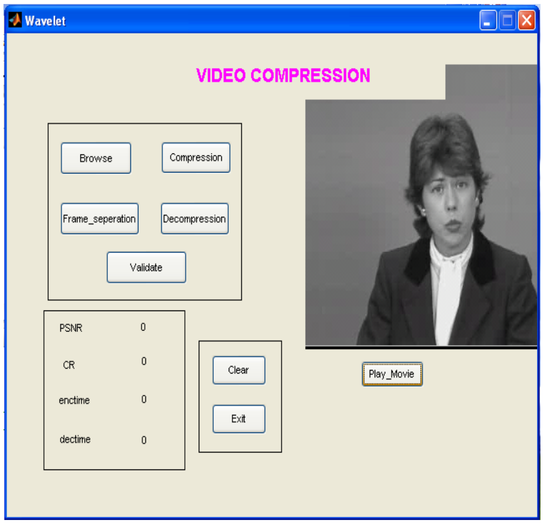
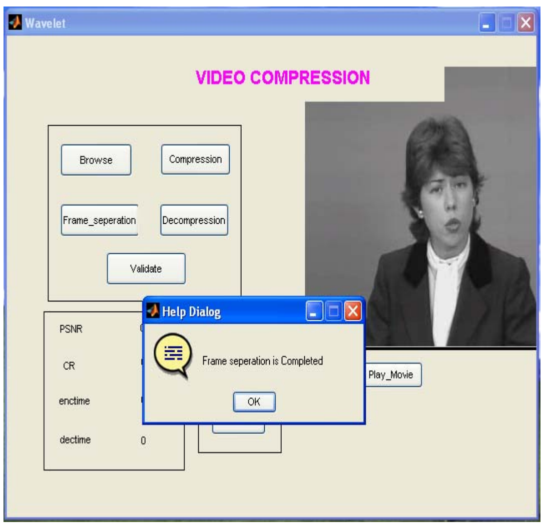
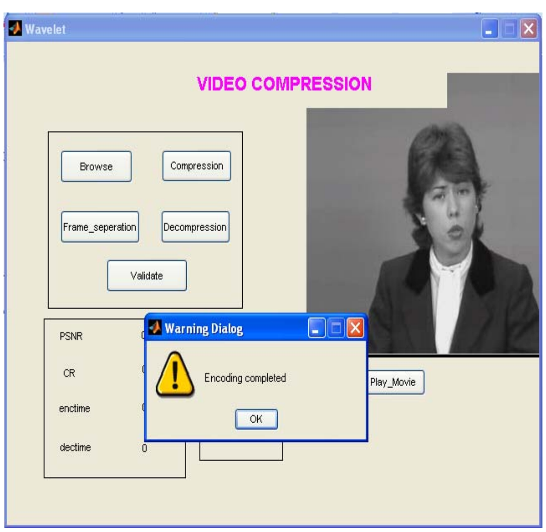


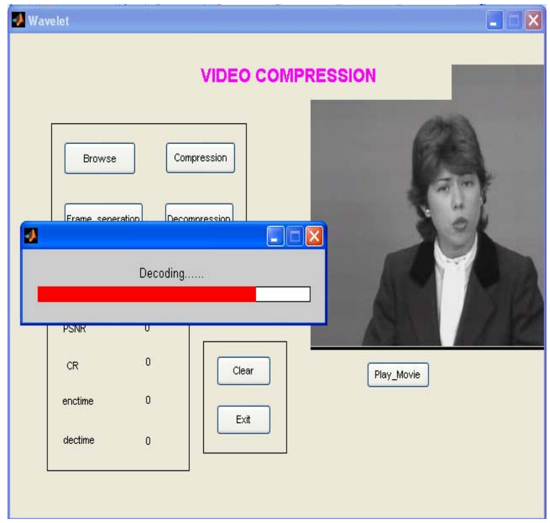
Figure 1:(a) GUI for video compression system



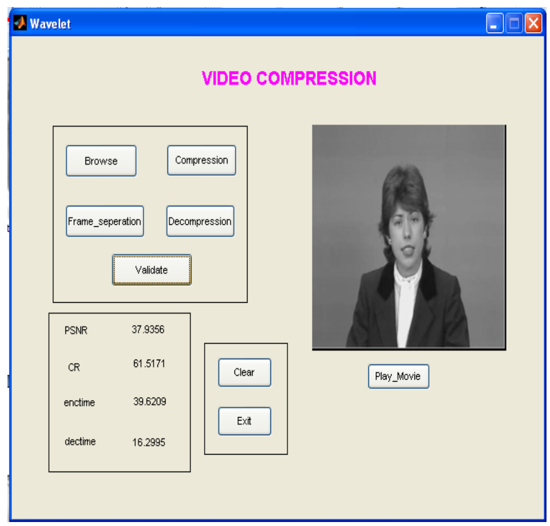
(b) Frame Separation



(c) Encoding Process



(d) Decoding Process



(e) Validation

Figure 2: Mother and daughter frame 92.

(a) Original

(b) LM SPIHT (PSNR=34.87dB)

(c) Proposed Method (PSNR=35.13dB)

Figure 3: Hall monitor frame 55. (a) Original

(b) LM SPIHT (PSNR=32.94dB)

(c) Proposed Method (PSNR=33.14dB)

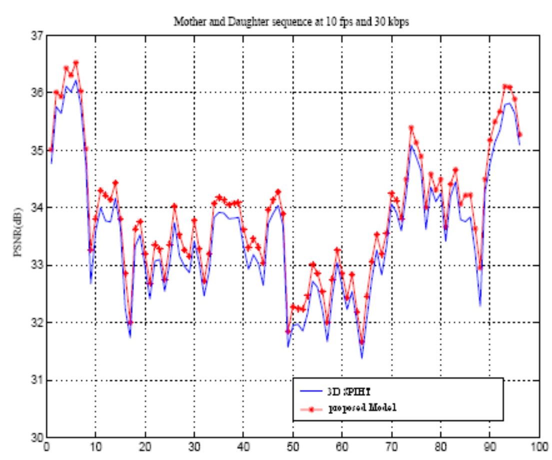
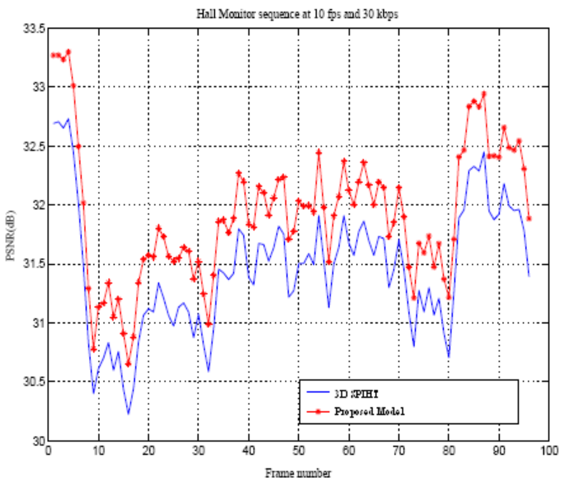


Figure 4: Frame Number Versus PSNR (a) Mother and Daughter sequence



(b)Hall monitor sequence

To conclude the performance betterment of our proposed method PSNR value is calculated, and then compared with two conventional methods shown in Table.1. This performance evaluated by PSNR (peak signal to noise ratio). PSNR value has been accepted as a widely used quality measurement in the field of image and video compression. To show the entire performance of our proposed algorithm, some of the simulated results were shown in this section. Here three different samples were discussed to check the robustness of the proposed method, such as Fig.1: News reader video sequence and its step by step process have been shown here like frame separation, encoding, decoding, finally validation. Fig.2: Mother and Daughter sequence, Fig.3: Hall monitor sequence, Fig 4 represented to show the calculated PSNR value for the above mentioned sequences, also for the every individual frames PSNR value is plotted.

Table.1. Comparison of PSNR for previous methods Vs proposed method

	3D SPIHT	**LM SPIHT**	**PROPOSED METHOD**
Mother and daughter	**34.62**	**34.87**	**35.13**
Hall monitor	**31.97**	**32.94**	**33.14**
News	**34.39**	**34.57**	**35.0243**
Walker	**32.64**	**33.82**	**34.2671**
Movie	**34.18**	**34.31**	**34.8558**

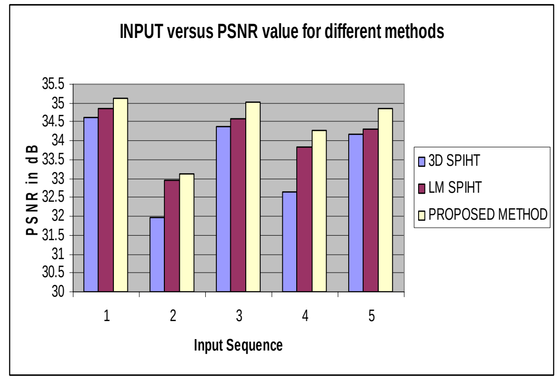


Figure 4: Sample versus Average PSNR value for all frames

From the above chart we can clearly differentiate the PSNR values for the five various samples with 3D SPIHT, LM SPIHT and proposed method. This evidence concludes that our method is works well than other methods.

5. CONCLUSION

In this paper, we successfully extended and implemented an adaptive scalable hierarchical tree algorithm on MATLAB and provided experimental results to show that our method is better than the previous SPIHT models. We not only improved the coding efficiency in the proposed encoding algorithm but also reduced memory utilization; significant computational time saving can also be obtained with the multilayered decoding scheme. In addition, spatial and temporal scalability can be easily incorporated into the system to meet various types of display parameter requirements and self-adapting rate allocations are automatically achieved. Finally our experimental results show that these methods can truly work well without much degradation in performance. Future research could concentrate on optimizing error rate, realizing the implementation on hardware.

REFERENCES

[1]. A.Said and W. Pearlman, “A New,Fast and Efficient Image Code Based on Set Partitioning in Hierarchical Trees ” IEEE Transactions on Circuits and Systems for Video Technology, vol.63, No.3, pp.243-250, June 1996.

[2]. J.M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients.” IEEE Transactions on Signal Processing , Vol. 41, No.12, December

[3]. B. J. Kim, Z. Xiong, and W. A. Pearlman, "Low Bit-Rate Scalable Video Coding with3D Set Partitioning in Hierarchical Trees (3-D SPIHT)." In IEEE Trans. On Circuits and Systems for Video Technology, Dec. 2000.

[4]. B.J. Kim and Pearlman, W.A.,"An Embedded Wavelet Video Coder Using Three- Dimensional Set Partitioning in Hierarchical Trees (SPIHT)" Proc. Data Compression Conference, 1997, pp.251-260, Snowbird, UT, USA, March 1997.

[5]. Yong Sun, Hui Zhang and Guangshu Hu."Real-Time Implementation of a New Leow- Memory SPIHT Image Coding Algorithm Using DSP Chip" IEEE Transactions on Image Processing, vol.11 No.9, Sept 2002.

[6]. Ekram Khan and Md. Ghanbari " Video Coding with Virtual Set Partitioning in Hierarchical tree " IEEE Transactions on Signal Processing, vol.1 pp 449-452, 2002

[7]. Wen-ChienYen and Yen-Yu Chen "Natural Image Compression Based on Modified SPIHT" Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05) 0-7695-2296-3/05 © 2005 IEEE.

[8]. Shaorong Chang and Lawrence Carin "A Modified SPIHT Algorithm for Image Coding With a Joint MSE and Classification Distortion Measure " IEEE Trans.Image Processing, vol.15, no .3, march 2006, pp.713-725.

[9]. W. A. Pearlman, B.-J . Kim, and Z. Xiong,"Embedded video subband coding with 3D SPIHT," in Wavelet Image and Video Compression, P.Topiwala,Ed. Norwell, MA: Kluwer, 1998.

[10]. A. Wang, Z. Xiong, P. A. Chou, and S. Mehrotra, "Three- dimensional wavelet coding of video with global motion compensation," in Proc.Data Compression Conf.,Snowbird,UT,Mar.1999, pp.404–413.

[11]. Peter Symes "Video Compression Demystified" Megraw-Hill, 2001.

[12]. Iain E.G.Richardson,"H.264 and MPEG-4 Video Compression" John Wiley & Sons Ltd, 2003.

[13]. Jun Ho Cho, Tae Gyoung Ahn, and Jae Hun Lee "Lossless Image/Video Compression with Adaptive Plane Split" IEEE Transactions on Signal Processing, 978-1-4244-2559-4, 2009.

[14]. Ali Aghagolzadeh, Saeed Meshgini, Mehdi Nooshyar, and mehdi Aghagozadeh"A Novel Video Compression Technique for Very Low Bit-Rate Coding by Combining H.264/AVC Standard and 2-D Wavelet Transform", ICSP Proceeding, pp1251-1254, 2008.