Color Image Coding Using a Novel Block Based Embedded Technique

Ranjan K. Senapati
Dept. of ECE, K L University, Green Fields
Vaddeswaram, Guntur (Dist.), Andhra Pradesh, India, 522502
Email: ranjan.senapati@kluniversity.in

P. M. K. Prasad
Dept. of ECE, GMR Institute of Technology, Rajam
Srikakulum (Dist.), Andhra Pradesh, India, 532127
Email: mkprasad.p@gmrit.org

Abstract—In this paper, an efficient, low memory embedded wavelet based color image coding technique is presented. The proposed algorithm is a listless variant of improved color set partitioning embedded block coding (CSPECK) algorithm. The inter-dependency among color planes is exploited by using novel clustering arrangement of luminance and chrominance subbands. This facilitate to encode simultaneously inter and intra subbands of luminance and chrominance components located at the same spatial locations in a hierarchical wavelet arrangement. This technique shows a significant improvement of peak-signal-to-noise-ratio (PSNR) values in most of the standard color images and videos at lower bit rates. The memory requirement of the coder is reduced by almost 84% than CSPECK.

Index Terms—Image compression, Embedded coder, Set partitioning in hierarchical trees, Set partitioning embedded block, Peak signal to noise ratio.

I. INTRODUCTION

Most of the wavelet based coders have achieved efficient compression by exploiting both inter and intra subband correlations through the use of either zero-tree or zero-block data structures respectively. The color set partitioning in embedded trees (CSPIHT) algorithm [1] has excellent rate distortion performance with low computational complexity. However, being a tree based algorithm, it is memory intensive. Color set partitioning embedded block (CSPECK) [2] is a block based fully embedded color image coding technique with low memory requirements. The algorithm proposed by Moiuddin et al. [3] shows a significant improvement in coding efficiency by integrating the set partitioning strategies of hierarchical trees and blocks into a single algorithm. However, the decoder is more complex than CSPIHT because of the use of additional step of quad partitioning. Further, the use of list data structures in these coders causes a variable data dependency memory management as the list nodes are updated. At higher rates, there can be as many list nodes as that of coefficients. This feature makes these coders less suitable for hardware implementations. An attempt to reduce the hardware requirements by applying listless feature in to the SPIHT was made by Wheeler and Pearlman [4]. Similarly, Latte et al. [5] incorporated listless feature in to the SPECK coder. However, the inefficiency of the above algorithms is that they generate many zeros at early bit plane passes. This in turn, lowers the low bit rate performance of the coder because zeros contribute nothing to PSNR values. Improvement at lower bit rates is a desirable feature for transmission of images through a narrow bandwidth channel where a significant amount of information is to be transmitted at the outset. It is evident from the coefficients of transformed image statistics that on early bit plane passes, many subband coefficients are typically insignificant below top one to six threshold level. The proposed algorithm encodes these large clusters of zeros at earlier passes more efficiently than CSPIHT, CSPECK and algorithm proposed in [3] to achieve enhanced low bit rate performance. Further, the proposed coder makes use of the advantage of listless implementation to achieve low memory requirements. Similar kind of idea is proposed by us in our previous papers [6], [7], [8]. The difference presented here is that the idea is extended to color images.

II. THE PROPOSED ALGORITHM

The proposed algorithm uses linear indexing scheme [4] to address a coefficient. There are three passes per bit plane. First, the insignificant pixel (IP) pass tests a lone significant pixel for significance (Pass 1). Secondly, the insignificant set (IS) pass tests each multiple pixel sets for significance (Pass 2). Initially, the IS pass, tests the significance of a combined wavelet level. A combined wavelet level consists of subbands of luminance and chrominance planes situated in the same spatial locations in an one dimensional array. If the combined level is insignificant with respect to the current threshold, the entire level can be skipped at once. This situation arises at top 1 to 5 bit planes in smooth images. For textured images it is up to 1-3 bit planes. The proposed composite level is block based. Whereas, the combined levels presented in [9] are tree based. Finally, in the refinement pass (Pass 3), the pixel which is found significant in the previous pass is refined. The partitioning rules of the proposed algorithm are presented below:

The proposed algorithm uses two types of state table markers. These are fixed marker $MF[k]$ and variable marker $MV[k]$. The values of $k$ are starting indices of each wavelet levels calculated in a zigzag-scan manner. Each marker holds 4 bits per coefficient to keep track of set partitions. For an image of $(N \times N)$ and $L$ levels of decompositions, the initial value of $MF[k]$ or $MV[k]$ is $\lfloor \log_2 N \rfloor - L$. The final value is $\log_2 P$. The markers in $MF[k]$ state keep track of wavelet levels rather than subbands in a wavelet level.
Y-plane

(a)

YCbCr coarsest level

YCbCr combined subband

CB-Plane

Cr-plane

(b)

So a particular pyramid level can be skipped at once by just encoding a single zero instead of 9 zeros as that of CSPECK. Thus, the gain is expected to increase by 9 times than CSPECK on the top luminance and chrominance planes. Because, for three color planes, $3 \times 3 = 9$ subbands to be encoded by 9 zeros in CSPECK. $MV[k]$ state table markers keep track of set partitions within a wavelet level. At any given time, if a wavelet level is significant with respect to a threshold, $MV[k]$ markers are updated by splitting the wavelet level. Like CSPECK, the proposed algorithm uses strictly breadth first search. The coefficients of YCbCr color planes are arranged in a one dimensional arrangement of length $I'_L$. The algorithm scans for each bit plane pass $n$. The significance level for each bit plane is $s = 2^n$, which is done with bitwise AND operation. In each pass, the coefficient array 'val' is examined for significance with respect to current threshold. The pseudo code for each bit plane pass of the proposed algorithm is presented in sub section C.

Considering YCbCr color space (4:2:0 format), each plane is separately wavelet transformed using 9/7 CDF filter to the same number of decomposition levels. The Fig. 1(a) shows two level of wavelet decomposition for ease of illustration whereas, four level of decompositions are used in our simulations for Lena and News images. Wavelet subbands of Y, Cb and Cr color planes are interleaved using the sequential numbering pattern as shown in Fig. 1(a). Then, two dimensional subbands are mapped to one dimensional arrangement as in Fig. 1(b) so that the decoder can generate a fully embedded bit stream. At higher bit plane passes this arrangement facilitates to encode a single symbol to a composite level, unlike many symbols required to encode in CSPHT or CSPECK. The subsampling and up sampling is done using Lanczos2 reconstruction method both at encoder and decoder sides.

A. Initialization of markers in CLEBP algorithm

We will consider a $512 \times 512$ Lena color image. After interleaving the subbands as shown in Fig. 1, the markers are initialized as follows:

Each marker and its meaning is listed below:

- $S * P$: Each of these three symbols is used for a single pixel.
  - SIP: The pixel is insignificant or untested for this bit plane.
  - SNP: The pixel is newly significant so it will not be refined for this bit plane.
  - SSP: The pixel is significant and will be refined in this bit plane.

The following markers are used on the leading node of each lower level of pyramid. As the image is scanned, these markers indicate that the next level or subband or a block is insignificant.

1. Initialize the total 1-D array.
   $MV[1 : 393216] = SIP$, except at the leading indices of each combined subbands. Where 393216 is the total length of 4:2:0 RGB planes after mapping into an one dimensional array as shown in Fig. 1(b).

2. Fixed markers ($MF[k]$):
   - $MF[1]$: The pixel is the first index of combined coarsest level. This pixel along with all coefficients in the same level can be skipped.
   - $MF[1537]$: The pixel is the first index of combined level L. This pixel along with all coefficients in the same level can be skipped.
   - $MF[6145]$: The pixel is the first index at combined level $L-1$. This pixel along with all coefficients in the same level can be skipped.
   - ...
   - $MF[98305]$: This is the first index at the finest combined level. This pixel along with all coefficients in this level can be skipped.

3. Variable markers ($MV[k]$): The function of variable markers for a typical pyramid level ($L − 1$) is explained below. The same can be generalized for other levels.
   - $MV[6145] = MF[6145]$ indicates the entire combined wavelet level ($L − 1$) can be skipped.
• \(MV[6145]=MF[6145]-1\) indicates a combined wavelet level is split into 3 combined subband, or a combined subband in that wavelet level \((L-1)\) can be skipped.
• \(MV[6145]=MF[6145]-2\) indicates a combined subband is split into 6 parts, or \(1/6^{th}\) of a combined subband block in the wavelet level \(L-1\) can be skipped.
• \(MV[6145] = MF[6145]-3\) indicates a \(1/4^{th}\) of a block can be skipped.
• \(MV[6145]=0\), then block size equals to pixel size. The pixel is to be tested for significance.

The combined level and combined subband are shown in Fig. 1(b). \(k = 1537, 6145, 24577, 98305\) are the first indices, starting from resolution level 4 to level 1(finest resolution level), for a total of 4 levels of arrangement in a \(512 \times 512\) color image with 4:2:0 sampling format.

1. From Eq. 1, the initialization of \(MF[k]\) and \(MV[k]\) markers are defined below:
   • \(MF[1537, 3073, 4609] = MV[1537, 3073, 4609] = 6\) are the leading nodes of \(HL_4, LH_4\) and \(HH_4\) combined subbands.
   • \(MF[6145, 12289, 18433] = MV[6145, 12289, 18433] = 7\) are the leading nodes of \(HL_3, LH_3\) and \(HH_3\) combined subbands.
   • \(MF[98305, 196609, 294913] = MV[98305, 196609, 294913] = 9\) are the leading nodes of \(HL_1, LH_1\) and \(HH_1\) combined subbands.

3. \(MV[k]\) will be initialized to any value greater than highest marker value except for \(k \in \text{leading indices of each combined subbands. These } MV[k]\) markers are mark as \(SIP\).

B. Partitioning rule

The partitioning rule of a YCbCr combined level is shown in Fig. 2. In step 1, if any wavelet coefficient is found significant for a certain threshold, the entire YCbCr combined level is partitioned in to 3 parts. Each part is a YCbCr combined subband which is shown in step 2. Further, the combined subband is split in to 6 parts if any coefficient is found significant to the same threshold level as shown in step 3. Proceeding from step 4, each partitioned part recursively quad split to search for significant coefficient. Then, each significant coefficient is coded and transmitted along with the sign bit. The entire partitioning rule is presented in terms of Pseudocode in the following subsection-C.

C. Pseudocode of CLEBP Algorithm

Pass1: Insignificant pixel pass

1: Initialize \(k=0, k \leq I\)
2: if \(MV[k] \rightarrow \text{SIP then}\)
3: output(d = val[k] AND s)
4: if d == '1' then
5: send output(sign[k])
6: \(MV[k] \rightarrow \text{SNP}\)
7: else
8: move to next pixel
9: if
10: end if

Pass2: Insignificant set pass

1: if \(MF[k] = MV[k] \& (MV[k] \neq SIP)\) then
2: if \(k \in \text{Coarsest level then}\)
3: Test significance of level, send output('d')
4: if d=='1' then
5: Split into 6 parts.
6: else
7: skip to the next combined level.
8: end if
9: else if \(k \in \text{Combined level then}\)
10: Check significance of the combined level, send output('d')
11: if d=='1' then
12: Split the combined level into 3 parts
13: else
14: Skip the combined level
15: end if
16: if
17: else if \(MF[k] \neq MV[k] \& (MV[k] \neq SIP)\) then
18: send output('d')
19: if d=='1' then
20: Split into 6 parts
21: else
22: move to next combined subband
23: end if
24: else if \(MF[k] \neq MV[k] \& (MV[k] \neq SIP)\) then
25: if d=='1' then
26: Quadsplit the block
27: else
28: move past the block
29: end if
30: end if

Pass3: Refinement pass

1: if \(MV[k] \rightarrow \text{SSP then}\)
2: send output(d=val[k] AND s)
3: move past the pixel
4: else if \(MV[k] \rightarrow \text{SNP then}\)
5: \(MV[k] \rightarrow \text{SSP}\)
6: move past the pixel
7: else if check all the steps of insignificant set pass then
8: skip the block/set
9: else
10: skip pixel
11: end if

In pseudocode the following points are to be noted:
(MV[k] = MF[k] × (MV[k] ≠ SIP)) indicate YCbCr combined level is (Fig. 1(b)) to be tested for significance.

- (MV[k] = MF[k] − 1) ∧ (MV[k] ≠ SIP) indicate YCbCr combined subband is to be tested for significance.
- (MV[k] ≠ MF[k]) ∧ (MV[k] ≠ SIP) indicate a block within YCbCr combined subband is to be tested for significance.

### III. MEMORY REQUIREMENTS

Let R and C are number of rows and columns in luminance plane. L is the number of decomposition levels. In the proposed algorithm, coefficients are stored in one dimensional array of length \( 4L = (I + I/2) \). (Assuming the length of luminance plane, \( I = RC \). If \( W \) bits are needed for each subband coefficients, then the bulk memory required is \((I + \frac{1}{2})W \) for subband data; \( \frac{RC}{I} + \frac{RC}{4} \) for maximum descendant array; \( \frac{3L+1}{2} \) bytes for state table MV[k] (because, each marker needs half byte per pixel); \( \frac{3L+1}{2} \) bytes for MF[k] state table markers. The total memory required (4:2:0 YUV color plane format) by the proposed technique is: \( M(Proposed) = (I + \frac{1}{2})W + \frac{RC}{I} + \frac{RC}{4} + (3L + 1) \) bytes. This is all that the memory required by CLEBP including the whole image.

The memory required by CLEBP due to only marker state table is: \( \frac{RC}{I} + \frac{RC}{4} + (3L + 1) \) bytes = 192 kB. For CSPIHT, the memory required is: \( M(CSPIHT) = [W(N_{LIP} + N_{LIS} + N_{LSP}) + N_{LIS}] / 8 \) bytes. In worst case, \( N_{LIP} + N_{LSP} + N_{LIS} = [2 \times (M \times N)] + (M \times N) \) for three color planes. For CSPECK, the memory required is: \( M(CSPECK) = [W(N_{LIP} + N_{LIS} + N_{LSP})] / 8 \) bytes. For algorithm in [3], it is almost 90% of SPIHT, i.e., \( 0.9 \times M(CSPIHT) \) approx.

Therefore, the worst case memory required for a 512 × 512 image, with 18 bits per coefficients and 4 levels of coefficients arrangement; the proposed algorithm occupies 192 kB, CSPIHT occupies 1728 kB, CSPECK requires 1184 kB, and algorithm in [3] is 1560 kB. It is to be noted that the proposed algorithm takes a fixed amount (192 kB) whereas, the memory grows exponentially with each bit plane pass in CSPIHT,CSPECK and algorithm in [3].

### IV. SIMULATION RESULTS

In order to understand the effectiveness of the proposed algorithm, simulation is carried out on standard still color images, such as Lena of 512 × 512 size, first frame ‘News’ of common intermediate format (CIF: 352 × 288 size) of the standard MPEG-4 test video sequences. The comparison results are shown in Fig. 3 and Fig. 4. CLEBP(case 1) uses Lanczos2 subsampling and upsampling in chrominance planes, whereas CLEBP(case 2) subsamples chrominance planes by discarding alternate rows and columns. It is observed that PSNR values of CLEBP(case 1) outperform the results in [3] by 0.4-1.6 dB on Y plane and 0.7-2.8 dB on U-plane and 0.6-3.9 dB on V-planes. Considering CLEBP(case 2) in Lena image, higher performance is achieved in Y plane. But, the performance in chrominance planes goes down over 2 dB than other two algorithms. News image is compressed using slight different procedure than Lena image since its spatial dimensions are not equal. In order to make the dimensions
integer power of 2 (e.g. $512 \times 512$) the image is padded using border extension (replicate padding). The CLEBP algorithm that uses replicate padding and Lanzos2 interpolation is named as CLEBP(case 3). CLEBP using symmetric padding we called as CLEBP(case 4). Fig. 5 shows the a comparative result of CLEBP using replicate padding and symmetric padding. Therefore, using CLEBP(case 3), News image consistently outperforms results in [3] by 0.8-2.9 dB in Y-plane, 0.2-2.1 dB in U-plane and 0.3-2.4 dB in V-plane.

Summarizing the results it can be concluded that CLEBP shows a significant PSNR improvement at lower bit rates compared to CSPIHT and results in [3] than at higher bit rates.

V. CONCLUSIONS

An efficient, reduced memory listless embedded color image coding algorithm is presented. The algorithm improves low bit rate performance by efficiently encoding large clusters of zero blocks using few symbols compared to CSPIHT and algorithm by Moinuddin et al.. Compressing the bit string at lower rates also improve the overall performance on wide range of bit rates. The irregularities in chrominance planes depend on the subsampling and interpolation during image encoding and reconstruction. However, significant improvement in luminance plane at lower rates validate our assumption. Further, listless implementation technique reduces the memory requirements almost by 84% compared to algorithms using lists.

REFERENCES


