A VERY LOW BIT-RATE EMBEDDED COLOR IMAGE CODING WITH SPIHT

A Bouridane [*], F Khelifi **, A Amira*, F Kurugollu* and S Boussakta***

[*] School of Computer Science, Queen's University Belfast, United Kingdom. (e-mail: A.Bouridane@qub.ac.uk).

[**] Department of Electrical Engineering, University of Jijel, Algeria..

[***] School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, United Kingdom.

Abstract. In this paper, we propose an efficient extension of set partitioning in hierarchical trees (SPIHT) for very low bit-rate wavelet based color image coding. Since the chrominance components I and Q in YIQ format are sufficiently less significant in terms of energy compared to the luminance component, the trees within each chrominance plane are joined together in the list of insignificant sets (LIS) according to a virtual relationship parent-descendants specific to chrominance components. For generating fully embedded bit stream similar to SPIHT, the proposed method improves the performance of color SPIHT based scheme, especially for very low bit rate.

1. Introduction

The Discrete Wavelet Transform (DWT) has matured and has become very effective for the compression of natural images. Since the introduction of the zerotree wavelet image coding by Shapiro in [3], several coding algorithms have been developed. Said and Pearlman proposed in [2] an improved scheme, called Set Partitioning In Hierarchical Trees (SPIHT). Both the shemes rely on partial ordering of the wavelet coefficients by magnitude, followed by bit plane progressive refinement. The bitstream generated is perfectly embedded. For color image coding, an extension of 3D SPIHT has been proposed by Kim and Pearlman in [4] and applied to color video compression. The main idea of the color SPIHT consits of treating all color planes as one unit at the coding stage before generating one mixed bit-stream. Thus, the coding can stop at any point to give a precise bit-rate control.

For most natural multispectral images in NTSC YIQ color space, in addition to the redundancy within each subband of each component, the two chrominance components I and Q are significantly correlated in terms of insignificance of trees within each chrominance plane when compared with a given threshold. Indeed, the fixed threshold at the initialization stage with SPIHT depends only on the largest coefficient magnitude of luminance image because of the concentration of considerable fraction of image color energy in the luminance component. For very low bit rate color image coding using SPIHT, although this threshold decreases by half in each quantization step, a large number of trees in the same spatial orientation of each chrominance component are insignificant in comparison with the current threshold. Therefore, in order to reduce the number of bits required for coding the chrominance components and exploit this inter-component redundancy for very low bit-rate compression, we propose, in this paper, a virtual relationship parent-descendants with the aim of joining the two chrominance components in the significant test of trees.

This paper is organized as follow: the next section, section 2, describes in detail the proposed modification of SPIHT for color image coding. In section 3, we present the implementation details and an assessment of the results obtained with comparison with color SPIHT scheme. Section 4 concludes the paper.

2. Color image coding with SPIHT

In this section, a brief description of the SPIHT algorithm is first given, then our proposed method for color image coding is described. The SPIHT consists of three stages: initialization, sorting pass and refinement pass. It sorts the information of wavelet coefficients in three ordered lists: list of insignificant sets LIS, list of insignificant pixels LIP and list of significant pixels LSP. At the initialization stage the SPIHT first defines a start threshold according to the maximum value in the wavelet coefficients pyramid, then sets the LSP as an empty list and puts the coordinates of all coefficients in the coarsest level of the wavelet pyramid (LL band) in the LIP and those wich have descendants to the LIS. In the sorting pass, the algorithm first starts to sort the elements in the LIP then in the LIS. For each pixel in the LIP it performs a significance test against the current threshold and outputs the test result (0 or 1) to the output bitstream. If a coefficient is significant, its sign is coded and then its coordinate is moved to the LSP. During the sorting pass of LIS, the SPIHT does the significance test for each set in the LIS and outputs the significance information (0 or 1). If a set is significant, it is partitioned into its offspring and leaves. After the sorting pass for all elements in the LIP and LIS is done, SPIHT does a refinement pass with the current threshold for all entries in the LSP, except those which have been found significant during the last sorting pass. Then the current threshold is divided by 2 and the sorting and refinement stages are continued until we achieve the target bit-rate [2].

2.1. The proposed technique

The 24 bit color images used for our study contain three components: Red, Green and Blue (RGB), with eight bits being allocated to each color. Before wavelet decomposition, the RGB components are linearly transformed to the YIQ components. The Y component is the luminance image, and the I and Q components are the imphase and quadrature images. A simple application of SPIHT to color image would be to code each color plane separately. Then, the generated bit-stream of each plane would be serially concatenated. However, this simple method would require allocation of bits among color components thus losing precise rate control and would fail abandon the full embeddeness of the codec since the decoder needs to wait until the full bit-stream arrives to reconstruct the color image. Instead, Kim and Pearlman treated all color planes in [4] as one unit at the coding stage and generated one mixed bit-stream in such a way that we can stop at any point of the bit-stream and recontruct the color image of the best quality at the given bit-rate. In addition, the algorithm automatically allocates bits optimally among the color planes. By doing so, we will still keep the full embeddedness and precise rate color of SPIHT. After the separable decomposition of eah color plane (supposing in format YIQ), the Kim and Pearlman's method initializes the LIP and LIS with the appropriate coordinates of the top level in all three planes. Since each color plane has its own spatial orientation trees, it automatically assigns the bits among the planes according to the significance of the magnitudes of their own coordinates.

For most natural color images, in addition to the redundancy within each subband of each component, the chrominance components I and Q are significantly correlated in terms of insignificant trees when with a given threshold. In fact, the fixed threshold at initialization in the algorithm of SPIHT depends only on the largest coefficient magnitude of luminance image because of the concentration of considerable fraction of color image energy in the luminance component. For very low bit rate color image coding with SPIHT, although this threshold is divided by two in each quantization step, a large number of zerotrees in the same spatial orientation of each chrominance component remain insignificant [4].

The proposed modification consists of grouping the trees within each chrominance plane in the LIS during the sorting pass according to a virtual relationship parent-descendants specific to chrominance components. Fig. 1 shows how our virtual parent-descendants relationship is defined.

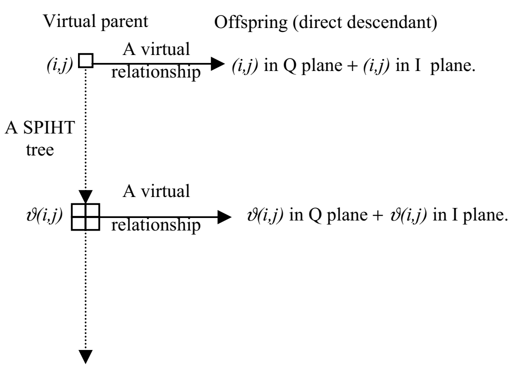


Figure 1 . Virtual dependencies parent-offspring in the chrominance domain.

Each virtual parent at coordinate *(i, j)* has two children (direct descendants) at the same coordinates in the two planes of chrominance I and Q. According to the SPIHT tree, its four offspring $\vartheta(i,j)$ have also four children in plane I and four children in plane Q at the corresponding coordinates. The following sets of coordinates are used to present our method:

• $\vartheta(i,j)$: set of coordinates of all offspring of node *(i, j)* as defined by SPIHT tree.

• $\vartheta c(i,j)$: set of coordinates of all chrominance offspring of virtual node *(i, j)*.

• *D(i, j)*: set of coordinates of all descendants of node *(i, j)* as defined by SPIHT tree.

• *Dc(i, j)*: set of coordinates of all descendants of node *(i, j)* in the virtual relationship sens of chrominance.

• *H(i, j)*: set of coordinates of all sptial orientation roots as defined by SPIHT dependencies.

• *Hc(i, j)*: set of coordinates of all pixels in the highest level.

• $L(i,j) = D(i,j) - \vartheta(i,j)$.

• $Lc(i,j) = Dc(i,j) - \vartheta c(i,j)$.

To clarify the relationship between magnitude comparisons and message bits, the function $S_n(\cdot)$ is used to identify significant coefficients or significant sets as defined in the following equation:

$$S_n(\Gamma) = \begin{cases} 1, & \text{if any } \left|C_{i,j}\right| \geq 2^n \text{ where } (i,j) \in \Gamma \\ 0, & \text{otherwise} \end{cases}$$

Proposed algorithm:

1) Initialization

–Output $n = \left\lfloor \log 2\left(\max_{(i,j)\in Y,I,Q}\left\{\left|C_{i,j}\right|\right\}\right)\right\rfloor$;

–Output $nc = \left\lfloor \log 2\left(\max_{(i,j)\in I,Q}\left\{\left|C_{i,j}\right|\right\}\right)\right\rfloor$;

–Set LSP as an empty list, add the coordinates $(i,j) \in H$ to the LIP with luminance symbol 'Y' and add only those with descendants also to LIS with luminance symbol 'ℓ' as type A entries.

–Add the coordinates $(i,j) \in Hc$ to the LIS with chrominance symbol '*c*' as type A entries.

2) Sorting pass

– for each entry *(i, j)* in the LIP with symbol component 'Y', 'I' or 'Q', output $S_n(i,j)$ in the corresponding color plane.

– if $S_n(i,j)=1$ then move *(i, j)* with corresponding component symbol to the LSP and output the sign of $C_{i,j}$ in the corresponding color plane.

– for each entry *(i, j)* in the LIS do

– if *(i, j)* is a luminance parent (with symbol 'ℓ' in the LIS):

– if the entry is of type A then:

• output $S_n(D(i,j))$;

• if $S_n(D(i,j))=1$ then:

∗ for each $(\kappa, \ell) \in \vartheta(i,j)$ do

• output $S_n(\kappa, \ell)$;

• if $S_n(\kappa, \ell) = 1$ then add (κ, ℓ) to the LSP with component symbol 'Y' and output the sign of $C_{\kappa,\ell}$.

• if $S_n(\kappa, \ell) = 0$ then add (κ, ℓ) to the end of the LIP with component symbol 'Y'.

∗ if $L(i,j) \neq \varnothing$ then move *(i, j)* to the end of LIS with component symbol 'ℓ' as an entry of type B. Otherwise, remove entry *(i, j)* from the LIS.

– if the entry is of type B then:

• output $S_n(L(i,j))$;

• if $S_n(L(i, j))$=1 then:

∗ add each $(\kappa, \ell) \in \vartheta(i, j)$ to the end of the LIS with component symbol 'ℓ' as an entry of type A.

∗ remove *(i, j)* from the LIS.

– if *(i, j)* is a virtual chrominance parent (with symbol '*c*' in the LIS) and nc ≥ n then:

– if the entry is of type A then:

• output $S_n(Dc(i, j))$;

• if $S_n(Dc(i, j))$=1 then:

∗ output $S_n(i, j) \in \vartheta c(i, j)$ in the plane I.

• if $S_n(i, j)$ =1 then add *(i, j)* to the LSP with component symbol 'I' and output the sign of $C_{i,j} \in$ plane I.

• if $S_n(i, j)$ =0 then add *(i, j)* to the end of the LIP with component symbol 'I'.

∗ output $S_n(i, j) \in \vartheta c(i, j)$ in the plane Q.

• if $S_n(i, j)$ =1 then add *(i, j)* to the LSP with component symbol'Q'and output the sign of $C_{i,j} \in$ plane Q.

• if $S_n(i, j)$ =0 then add *(i, j)* to the end of the LIP with component symbol 'Q'.

∗ if $Lc(i, j) \neq \emptyset$ then move *(i, j)* to the end of LIS with component symbol '*c*' as an entry of type B. Otherwise, remove entry *(i, j)* from the LIS.

– if the entry is of type B then:

• output $S_n(Lc(i, j))$;

• if $S_n(Lc(i, j))$=1 then:

∗ add each $(\kappa, \ell) \in \vartheta(i, j)$ to the end of the LIS with component symbol '*c*' as an entry of type A.

∗ remove *(i, j)* from the LIS.

3) Refinement pass

– for each entry *(i, j)* in the LSP with symbol component 'Y', 'I' or 'Q', except those included in the last sorting pass, output the *n*th most significant bit of $|C_{i,j}|$.

4) Quantization step

– decrement *n* by 1 and go to step 2.

The initial value of '*n*' represents the number of bits of the largest coefficient among Y, I and Q components. The initial threshold for SPIHT coding is then chosen to be 2^n. It is important to note that the SPIHT algorithm is used in its entirety for the luminance plane in the sorting and refinement pass. However, the chrominance planes are joined together in the LIS at the sorting stage using the virtual relationship described above. In this case, the condition '$nc \geq n$' is added because the Y component contains more of the original image's information content than I and Q do. Therefore, at initialization stage, it is clear that all chrominance sets in the LIS are insignificant according to the fact that most of color image energy is packed into the luminance plane.

The task of the decoder is to replace the word 'output' by 'input' and to update the reconstructed color image. For the value of '*n*' when a coordinate is moved to the LSP, the decoder uses the magnitude according to the current threshold 2^n, plus the sign bit that is inputted after the insertion in the LSP, that's is to put $\hat{C}_{i,j} = \pm 1.5 \times 2^n$. On the other hand, during the refinement pass, the decoder adds or subtracts 2^{n-1} to all entries in the LSP, except those which have been moved to the LSP during the last sorting pass, in accordance with the bits of the binary representation of $|C_{i,j}|$ that inputs.

3. Implementation and results

Our proposed technique has been assessed against existing and similar methods and the following results were obtained using 24-bit color, 512×512 images downloaded from the web site [5]. Although, all images were used in the analysis, only the girl (Tiffany) (image No. 4.2.2), Lena (4.2.4) and Peppers (4.2.7) are shown in this paper. First, the images were linearly transformed to the YIQ color space. Then, a 5-stage dyadic wavelet decomposition of each color component was carried out using the 9/7-tap bi-orthogonal filters of [1]. The distortion is measured by the overall PSNR which was computed from the combined mean-squared-error of the red, green, and blue components after the reconstruction.

Table 1 shows a comparative evaluation of the results obtained for three images with three different bit rates.

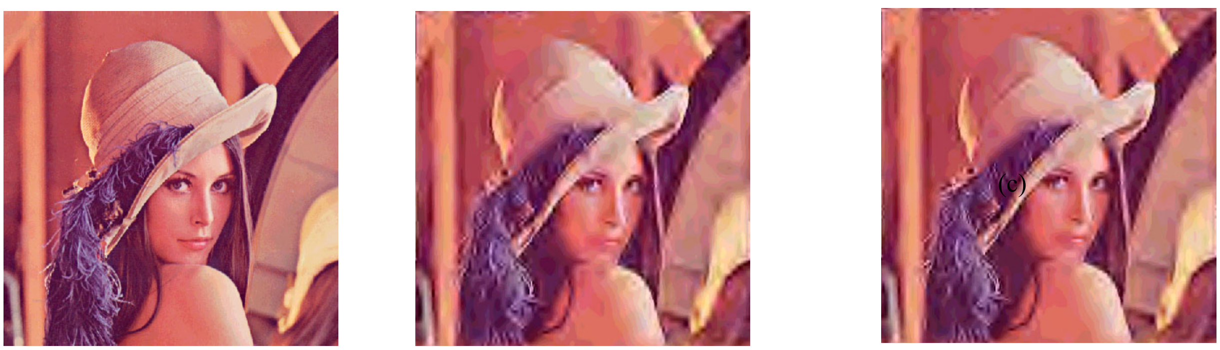
bpp	Separated method	Kim and pearlman's method	Proposed method
0.03125	23.86 dB	22.88 dB	24.02 dB
	22.22 dB	21.62 dB	22.86 dB
	20.70 dB	19.99 dB	20.67 dB
0.0625	25.53 dB	25.63 dB	26.05 dB
	24.24 dB	24.55 dB	25.16 dB
	22.95 dB	22.52 dB	23.00 dB
0.125	27.95 dB	28.07 dB	28.29 dB
	26.27 dB	27.15 dB	27.36 dB
	25.12 dB	25.00 dB	25.15 dB

Table 1. Statistical results obtained

The analysis was performed at different very low bit rates ranging from 0.03 to 0.15 bit per pixel (bpp). The overall PSNR results obtained by our method against those proposed by Kim and Pearlman's method [4] and SPIHT coding of the red, green, and blue components separately, with equal bit allocation. These results show that at very low bit-rates, for all reconstructed images, the PSNR value for the proposed method outperforms the PSNR value of the Kim and Pearlman's method. The differences further increase as the bit-rate decreases. Figures 2 and 3 show the original images and their reconstructed versions at bit-rate of 0.0625 bpp. In these examples, especially on some edges, we can easily see a better quality provided by the proposed method.

(a) (b) (c)

Figure 2. Peppers at 0.0625 bpp. (a) original. (b) Kim & Pearlman's method (PSNR=22.52 dB). (c) Proposed method (PSNR=23.00 dB).



(a) (b) (c)

Figure 3. Lenna at 0.0625 bpp. (a) original. (b) Kim & Pearlman's method (PSNR=24.55 dB). (c) Proposed method (PSNR=25.16 dB).

4. Conclusion

A very low bit-rate embedded color image compression scheme based on the wavelet SPIHT methodology has been developed. Our coding results outperform the color SPIHT algorithm introduced by Kim and Pearlman in [4]. The coded images present a subjective quality that is satisfactory for a large number of applications. Like the color SPIHT, the proposed method keep the full embeddedness and the progressive transmission of the coding scheme. Also using an arithmetic coding on output bit stream can improve the PSNR.

5. References

[1] M.Antonini, M.Barlaud, P.Mathieu and I.Daubechies, " Image coding using wavelet transform, " *IEEE Trans. Image Processing*, Vol. 1, pp. 205-220, April 1992.

[2] 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., pp. 243-250, June 1996.

[3] J.M.Shapiro, " Embedded image coding using zerotrees of wavelet coefficients, " *IEEE Trans. Signal Processing*, Vol. 41, pp. 3445-3462, Dec. 1993.

[4] B-J. Kim, Z. Xiong and W.A. Pearlman, " Low Bit Rate Scalable Video Coding With 3D Set Partitionig in Hierarchical Trees (3D SPIHT)," IEEE Trans. Circuits and Systems for video Technology, Vol. 10, pp.1374-1387, Dec. 2000.

[5] University of Southern California, USC image database at http://sipi.usc.edu/services/database.