EFFICIENT LOSSLESS COLOUR IMAGE CODING WITH SPECK

F. Khelifi, A. Bouridane and F. Kurugollu

School of computer science
Queen's University Belfast, Belfast, UK
phone: +44(0)28 9097 4860. Fax: +44 (0)28 9097 5666
{fkhelifi01, a.bouridane, f.kurugollu}@qub.ac.uk

ABSTRACT

This paper proposes an efficient extension of Set Partitioning Embedded bloCK (SPECK) algorithm to lossless colour image coding by using the integer wavelet transform. First, the RGB image is losslessly transformed to LC (Luminance-Chrominance) plane. Then, an integer wavelet transform is applied to each plane. Depending on the energy of each transformed plane and the correlation between each pair of planes, we select two planes to be grouped together into the List of Insignificant Sets LIS in SPECK algorithm in order to exploit the inter-redundancy of information so as to achieve a better performance of coding. The idea behind this is that the sets in LIS at the same location in two correlated planes with close energy are very likely to have the same information of significance with respect to a given threshold. Hence, joining them together can yield an important gain of the amount of bits. This novel method has been assessed in comparison to the separated one presented in [5] and the simulation results show a better performance of the proposed technique.

2. INTRODUCTION

In recent years, with the advanced development in internet and multimedia technology, the amount of image information that is broadcast by computers has grown and become more challenging. Indeed, this information requires a large amount of storage space and transmission bandwidth. Image compression, which is the technology of image data reduction, offers a valuable solution to save storage space and transmission bandwidth. Through the last decade, the discret wavelet transform (DWT) has matured and has become very effective for the compression of digital images. Since the introduction of the zerotree wavelet image coding by Shapiro in [1], several wavelet-based coding algorithms have been developed. Said and Pearlman proposed an improved scheme, called Set Partitioning In Hierarchical Trees (SPIHT) [2]. Both schemes rely on partial ordering of the wavelet coefficients by magnitude, followed by bit plane progressive refinement, but they differ in terms of partitioning mechanism. On the other hand, Islam and Pearlman proposed an embedded block-based compression technique, called SPECK, of lower complexity [3]. Exhibiting the same properties of SPIHT, SPECK yields a comparable performance to SPIHT with a faster encoding/decoding. All these techniques are primarily designed for gray-scale images and applied for lossy compression. However, in some applications, especially when the original database is required for further processing, such as image classification and recognition in medical field, the need of lossless coding arises as a necessity. Pearlman *et al* have shown that SPECK is fairly comparable to the best known compression techniques when applied for lossless coding [4]. In this paper, lossless colour image coding with SPECK is addressed. For coding of colour images, each colour plane can be treated separately from the others [4]. However, this would not exploit the inter-redundancy between the correlated transformed planes. In fact, two correlated planes with close energy yield similar information of coding in terms of signification when compared to a given threshold. Therefore, grouping them in the LIS can significantly reduce the number of bits required for coding. Starting from this idea, an efficient extension of SPECK to lossless colour image coding is proposed. In the next section, the SPECK algorithm is briefly described. Then, a description of the proposed technique is given in section 3. Section 4 provides the implementation details and simulation results. Finally, a conclusion is presented in section 5.

2. SPECK ALGORITHM

Similar to SPIHT, SPECK consists of three stages: initialisation, sorting and refinement. However, it sorts the information of wavelet coefficients in two ordered lists. List of Insignificant Sets LIS and the List of Significant Pixels LSP. At the initialisation stage, a start threshold depending of the maximum value in the wavelet coefficients pyramid is defined (chosen as a power of two: $T=2^{(n-1)}$) . The list LSP is set as empty. Then, the image is portioned into two groups (see Fig. 1): set of type S which is the root of the pyramid, and set of type Γ which is the rest of the image. From the standpoint of implementation, a block of set of type S is determined by the coordinates of the pixel in the top-left and the size of this block. First, set of type S is added to the LIS. In the sorting pass, the algorithm first starts to sort each block of type S in LIS by performing a significance test against the current threshold (1 or 0). A block is said significant if there is at least one coefficient in this block whose magnitude is greater than or equals the threshold. If a block of type S is significant, it is partitioned into four subsets of the same type (S_0, S_1, $S2$ and S_3) as shown in Figure.2.

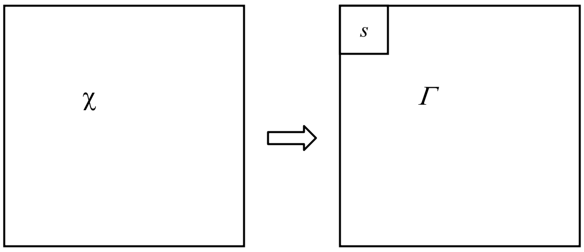

Fig. 1. Partitioning of image χ into stets S and Γ.

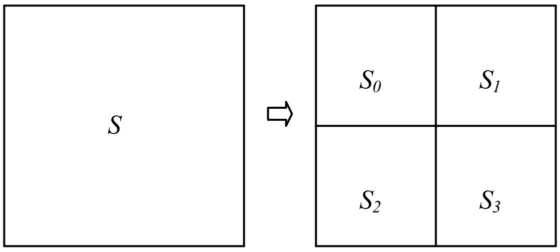

Fig. 2. Partitioning of stet S.

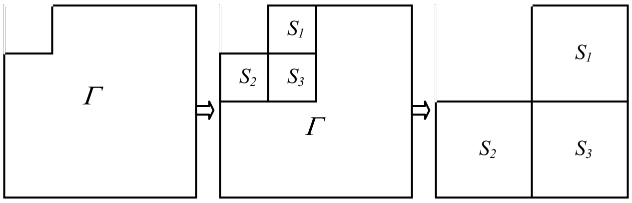

Fig. 3. Partitioning of stet Γ.

In the LIS, this block is replaced by the resulting subsets. In the case of a significant block of size 1×1 (one pixel), its sign is coded and then its coordinate is moved to the LSP. In the same way, the set Γ is tested with respect to the current threshold where its split up produces one subset Γ and three subsets of type S (S_1, $S2$and S_3) as depicted in Figure 3. This significance and partitioning process is carried out for all sets of type S (including the new ones) and the set Γ. Note that, depending upon the information content of the image and the desired bit-rate of coding, the set Γ can disappear at a certain point. In the refinement pass, the n-th most significant bit of each entry in the LSP, excluding those which have been added during the last sorting pass, is output. Then, the current threshold is divided by 2 and the sorting and refinement stages are continued until a predefined bit-rate is achieved (lossy case) or a smallest value of n (n=1 lossless case).

2. APPLICATION TO LOSSLESS COLOUR IMAGES

2.1 RGB to LC transformation

Usually a full colour image is made up of three primary colours: red, green, and blue. In most cases, there is a strong correlation between the red, green, and blue components of the colour image. However, this colour space is not optimal for lossless compression since the energy is distributed over these planes. Reversible RGB to LC transformations reduce the correlation between the R, G and B components and pack the energy of the colour image into one component. Consequently, the most important information of the image is represented by a few coefficients. The most widely used transformations are given in the following:

The forward transformation RGB to $Y'I'Q'$ [5] is given by:

$$Y' = \left\lfloor \frac{\left\lfloor \frac{R+B}{2} \right\rfloor + G}{2} \right\rfloor, \qquad I' = R - B\,, \qquad Q' = \left\lfloor \frac{R+B}{2} \right\rfloor - G$$

where $\lfloor \cdot \rfloor$ denotes the floor function. The inverse transformation is defined as :

$$R = Y' + \left\lfloor \frac{Q'+1}{2} \right\rfloor + \left\lfloor \frac{I'+1}{2} \right\rfloor, \quad G = Y' - \left\lfloor \frac{Q'}{2} \right\rfloor,$$

$$B = Y' + \left\lfloor \frac{Q'+1}{2} \right\rfloor - \left\lfloor \frac{I'}{2} \right\rfloor$$

The RGB to $O_1O_2O_3$ forward and inverse transformations [6] are given as:

$$O_1 = \left\lfloor \frac{R+G+B}{3} + \frac{1}{2} \right\rfloor, \quad O_2 = \left\lfloor \frac{R-B}{2} + \frac{1}{2} \right\rfloor,$$

$$O_3 = B - 2G + R$$

where the inverse transformation is:

$$R = O_1 - O_2 + \left\lfloor \frac{O_3}{2} + \frac{1}{2} \right\rfloor - \left\lfloor \frac{O_3}{3} + \frac{1}{2} \right\rfloor, \quad G = O_1 - \left\lfloor \frac{O_3}{3} + \frac{1}{2} \right\rfloor,$$

$$B = O_1 + O_2 + O_3 - \left\lfloor \frac{O_3}{2} + \frac{1}{2} \right\rfloor - \left\lfloor \frac{O_3}{3} + \frac{1}{2} \right\rfloor$$

The lossless RGB to $Y_LU_LV_L$ transform [7]:

$$Y_L = G + \lfloor (0.299/0.587) \cdot R + (0.114/0.587) \cdot B \rfloor,$$

$$U_L = B - \lfloor 0.587 \cdot Y_L \rfloor$$

$$V_L = R - \lfloor 0.587 \cdot Y_L \rfloor$$

where the inverse transformation is as follows:

$$R = V_L + \lfloor 0.587 \cdot Y_L \rfloor, B = U_L + \lfloor 0.587 \cdot Y_L \rfloor,$$

$$G = Y_L - \lfloor (0.299/0.587) \cdot R + (0.114/0.587) \cdot B \rfloor$$

Finally, the RGB to YD_bD_r reversible transform [8] is:

$$Y = \left\lfloor \frac{R + 2G + B}{4} \right\rfloor, \; D_b = B - G$$

$$D_r = R - G$$

Thus, the RGB components are given by:

$$G = Y - \left\lfloor \frac{D_b + D_r}{4} \right\rfloor, \; B = D_b + G, \; R = D_r + G,$$

2.2 SPECK for colour images

A simple application of SPECK to colour images would be to code each colour space separately as does a conventional coder. In lossy coding applications, Pearlman *et al* [4] suggested the use of a common LSP for three colour planes while each colour space has its own LIS. The aim behind this idea is to keep the embeddedness of the algorithm and ensure an adequate bit allocation among colour components. However, for lossless coding, this technique yields the same performance as the separated coding. Although the colour transformation RGB to LC reduces the correlation between the components and packs the energy of the image into one plane, the best exploitation of the correlation can be additionally achieved with the transformation by combining similar planes/correlated in the SPECK algorithm.

To achieve more efficiency, the closest planes must be joined together in the significance test of blocks. It is worth noting that the similarity between two transformed planes is not merely related to the correlation between them, but also to the ratio of their energy since the significance test is carried out with the same threshold for both. Here, we give an experimental criterion to select two closest planes of the transformed colour image. Let X_i , i=1,..3 three planes obtained after integer wavelet transform and lossless RGB/LC colour transform. We first compute the energy W_i of each plane X_i . Then, the 2D correlation between each pair of planes with absolute values is computed. Let $C_{i,j}$ the correlation between $abs(X_i)$ and $abs(X_j)$ $(i \neq j)$. The two planes which provide the largest value of λ which is computed as follows:

$$\lambda = \frac{min(W_i \ , W_j \ )}{max(W_i \ , W_j \ )} C_{i,j}$$

are selected to be joined in the LIS in the course of SPECK algorithm. It is worth noting that the closest integer-wavelet transformed planes provide a large number of coefficients with same locations which can be coded together in the initial iterations. Table 1. shows, for various images, the number of coefficients at the same location in the two closest planes giving the same significance information with respect to the threshold.

From the viewpoint of implementation, another LIS is used for the grouped planes. In the sorting pass, a block is significant if there is at least a significant coefficient in the joined planes. In case of a block of size 1×1, the corresponding coefficients in the joined planes are coded and the sign of the significant one which is moved to the LSP is output.

3. SIMULATION RESULTS

The following results were obtained with 24-bit colour, 256×256 images downloaded from [9][10]. First, the images are transformed to a reversible colour space. It is important to note that for all images, the proposed technique has been applied with RGB to YD_bD_r transformation which yields the best results when compared to the other ones. Then, a 5-stage dyadic wavelet decomposition of each colour component is carried out using various integer wavelet transforms. The criterion used for evaluation is the bit rate. Table 2 shows the results obtained in comparison to those provided by Pearlman's method [4] which separately codes the three colour planes. It can be shown that the proposed method outperforms the separated one for all test images.

Table. 1 Redundancy of coding the two closest planes in YUV plane [6] with the integer wavelet transform (2,2)

	Rate of coefficients with similar significance information (%)				
Threshold	Lena	Barche	Peppers	Couple	Tulips
16	98.24	99.90	98.02	99.62	95.99
8	92.03	98.48	95.85	96.66	89.87
4	71.28	91.32	90.81	84.09	79.30
2	37.66	76.21	74.71	61.77	61.64
1	16.26	47.75	43.34	38.90	32.82

4. CONCLUSION

In this paper, an efficient extension of SPECK to lossless colour image coding has been proposed. In addition to apply the reversible transformation RGB to LC and the integer wavelet transform so as to exploit the inter and intra-redundancy of colour components, the two closest planes in terms of signification are joined together in SPECK algorithm in order to achieve more efficiency. The proposed technique has been assessed in comparison with the separated coding of each colour plane by using SPECK. The simulation results carried out on various colour images have shown a better performance of the proposed technique.

REFERENCES

[1] J.M Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, No.12, pp.3445–3462, Dec. 1993.

[2] A. Said and W.A Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees, " *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.6, No.3, pp.243–250, June 1996.

[3] A. Islam and W. A. Pearlman, " An embedded and efficient low-complexity hierarchical image coder," in *Proc. SPIE Visual Communications and Image Processing* , vol. 3653, Jan. 1999, pp. 294–305.

[4] W.A Pearlman, A. Islam, N. Nagaraj and A. Said, "Efficient, low-complexity image coding a set partitioning embedded block coder, " *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.14, No.11, pp.1219–1239, Nov. 2004.

[5] I. Singh, P. Agathoklis and A. Antoniou, "Lossless compression of color images using an improved integer based nonlinear wavelet transform, " *IEEE International Symposium on Circuits and Systems*, June 9-12, 1997, Hong Kong.

[6] T. Nakachi, T. Fujii and J. Suzuki, "Lossless and near lossless compression of still color images, " *Proc. of the International Conference on Image Processing* , Vol.1, pp.453-457, 1999.

[7] R.J. Van der Vleuten, "Low-complexity lossless and fine-granularity scalable near-lossless compression of color images, " *Data Compression Conference, 2002. Proceedings*, DCC 2002, 2-4 April 2002, pp.477.
[8] D. Taubman and M. *Marcellin,* " *JPEG2000: Image Compression Fundamentals*, " Kluwer Academic, Boston, 2002.
[9] http://decsai.ugr.es/cvg/dbimagenes/c256.php
[10] http://sipi.usc.edu/services/database/

Table 2. Lossless performance comparison (bit-rate bpp) for various integer wavelet transforms operating on components of different reversible colour transformations

		(2,2)	(2,4)	(4,2)	S+P
Lena	Separated $Y'I'Q'$	15.2899	15.2977	15.3352	15.5191
	Separated $O_1O_2O_3$	15.2233	15.2767	15.3141	15.4928
	Separated $Y_LU_LV_L$	15.3681	15.4452	15.4830	15.6637
	Separated YD_bD_r	15.3006	15.3412	15.3788	15.5714
	Proposed YD_bD_r	**15.1927**	**15.2446**	**15.2819**	**15.4616**
Barche	Separated $Y'I'Q'$	12.7245	12.8183	12.8497	13.0052
	Separated $O_1O_2O_3$	12.7573	12.7895	12.8208	12.9831
	Separated $Y_LU_LV_L$	12.8400	12.9528	12.9845	13.1264
	Separated YD_bD_r	12.7491	12.7943	12.8256	12.9748
	Proposed YD_bD_r	**12.7188**	**12.7609**	**12.7922**	**12.9439**
Peppers	Separated $Y'I'Q'$	13.1343	13.1845	13.2259	13.3839
	Separated $O_1O_2O_3$	13.1288	13.1831	13.1944	13.3612
	Separated $Y_LU_LV_L$	13.3130	13.3743	13.3603	13.5086
	Separated YD_bD_r	13.0563	13.1125	13.1316	13.2874
	Proposed YD_bD_r	**13.0385**	**13.0879**	**13.1136**	**13.2693**
Couple	Separated $Y'I'Q'$	12.6771	12.7021	12.8363	12.9997
	Separated $O_1O_2O_3$	12.7519	12.7743	12.8158	12.9776
	Separated $Y_LU_LV_L$	12.7373	12.7606	12.9720	13.1208
	Separated YD_bD_r	12.5673	12.5967	12.6217	12.7897
	Proposed YD_bD_r	**12.5159**	**12.5364**	**12.5780**	**12.7374**
Tulips	Separated $Y'I'Q'$	15.4400	15.4999	15.5349	15.7192
	Separated $O_1O_2O_3$	15.4195	15.4734	15.5084	15.6924
	Separated $Y_LU_LV_L$	15.6623	15.7133	15.6796	15.8656
	Separated YD_bD_r	15.3426	15.4035	15.4311	15.6142
	Proposed YD_bD_r	**15.0583**	**15.1132**	**15.1451**	**15.3248**