Significant bit-plane clustering technique for JPEG2000 image coding

T.T. Lu and P.C. Chang

The bit-plane clustering technique is applied to high-energy code blocks to enhance the energy compaction by rearranging the column positions in these code blocks. The energy compaction effect can improve the coding efficiency of JPEG2000, which results in an improvement of 6.88% bit-rate reduction at 0.1 bpp on average over JPEG2000.

Introduction: The JPEG2000 standard [1] has shown better performance than the widely used JPEG standard [2]. Nevertheless, efforts to improve JPEG2000 never stop. Long *et al.* [3] modified the quantisation step-size selection schemes for the uniform scalar quantisation used in JPEG2000 to improve the lossy compression. Lian *et al.* [4] proposed two skipping methods applied to the embedded block coding to reduce JPEG2000 encoding time.

Embedded block coding with optimised truncation (EBCOT) is the core technique in JPEG2000. EBCOT defines three fractional bit-plane passes: the significance propagation, magnitude refinement and cleanup passes. Each coefficient bit in a bit-plane is coded in one of the three fractional bit-plane passes and the contexts are determined by the current coefficient and its neighbouring coefficients. EBCOT can achieve high coding efficiency using context-based adaptive arithmetic coding and rate distortion optimisation. The motivation of the proposed clustering method is to raise the proportion of the significance state, for which the bit value is 1, in the significance propagation pass. To achieve this, significant quantised coefficients are clustered together so that these significant bits in the bit-plane scan can be encoded in the significance propagation pass, and thus the coding efficiency of EBCOT is improved.

Algorithm: As in JPEG2000, the source image is decomposed into different frequency subbands by wavelet transform. These subbands are partitioned into code blocks, and each code block is entropy coded independently. A code block consists of bit-planes that are divided into stripes. Each stripe contains four rows of bits, and each group of four bits is formed as a column in a bit-plane. The bit-planes in a code block are scanned from the top stripe towards the bottom stripe and from the left-most column to the right-most column. The lowest frequency subband does not need bit-plane clustering because the energy distribution is already highly compacted. In high-frequency subbands, significant bit-plane clustering (SBPC) is applied only to the significant code blocks in each bit-plane scan. The rearranged positions are recorded as the side information. Then, all the code blocks are coded with EBCOT from the most significant bit-plane (MSB) to the least significant bit-plane (LSB). Note that for the significance/insignificance decisions the magnitude in this algorithm is represented by the minimum number of bits needed to represent the coefficient. This also simplifies the calculation. At the decoder, the process is the reverse of the encoder. The bit-plane clustering procedure is described as follows:

Step 1: Initialisation. This is to determine the magnitude threshold T_p, represented by the number of bits needed:

$$T_p = N - p \tag{1}$$

where p, $p = 0, 1, 2, \ldots,$ is the number of bit-plane layers that are investigated for clustering, and N is the number of bits that can represent the maximum magnitude over all frequency subbands except the lowest subband.

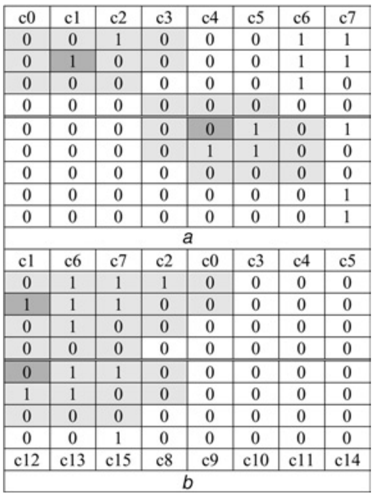
Step 2: Significant code block detection. If any of the coefficients is larger than the magnitude threshold, this code block is set as a significant code block. Go to step 3, significant code block rearranging. Otherwise, code block is insignificant and go to step 4.

Step 3: Significant code block rearranging and column clustering. This is to obtain the norm value for each column in significant code block. The norm $E[s][c]$, which is used to estimate the energy for simplicity, is then calculated as

$$E[s][c] = \sum_{i=0}^{3} \|q(u_s + i, v_c) \times F(u_s + i, v_c)\| \quad \forall s \in \{0, 1, \ldots, m-1\},$$
$$c \in \{0, 1, \ldots, w-1\} \tag{2}$$

where $q(u_s, v_c)$ represent the quantised coefficients in the code blocks, w is the width of the significant code block, and m is the number of stripes in the significant code block. If any of the coefficients $q(u_s, v_c)$ is larger than the magnitude threshold T_p, then the flag $q(u_s, v_c)$ is set to '1'. Otherwise, the flag is set to '0'. If any of the norm values are greater than zero in the current bit-plane scan and all the norm values at the same positions are zero in the previous bit-plane scans, the column is set as a new significant column. If the column is already rearranged in the previous bit-plane scan, the column is set as an old significant column. Otherwise, the column is insignificant. The new significant column with the maximum norm value is moved to the left of the stripe following the old significant columns and the other new significant columns are rearranged according to their norms, from high to low. The original positions of new significant columns need to be recorded. The positions of the old significant column are already recorded at the previous scan, and the insignificant columns are only shifted sequentially so that these positions need not be recorded in this bit-plane scan.

Step 4: Magnitude threshold update. If p is less than N, p is increased by one and go to step 2. Otherwise, the procedure stops and exits the algorithm.



c0	c1	c2	c3	c4	c5	c6	c7
0	0	1	0	0	0	1	1
0	1	0	0	0	0	1	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1
0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1

a

c1	c6	c7	c2	c0	c3	c4	c5
0	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
c12	c13	c15	c8	c9	c10	c11	c14

b

Fig. 1 *Sub-bit-plane coding order: (a) scanning result by using three coding passes without clustering, (b) scanning result with column rearranging*

Example: Fig. 1 shows an example of the sub-bit-plane coding order. A significant code block (size 8 × 8) contains two significant stripes, i.e. top four rows as one stripe and bottom four rows as another stripe, and each significant stripe is divided into eight columns, (c0, c1, …, c7) and (c8, c9, … c15). The dark areas, i.e. the second coefficient in c1 and the first coefficient in c12, are already significant in the previous bit-plane scan. Fig. 1*a* shows the result after three coding passes without column rearranging. Twenty-three coefficients, i.e. the top stripe has 13 and the bottom stripe has 10, are coded in the significance propagation pass as shown in the dark areas. Among them only four coefficients, the first coefficient in c2, the second coefficient in c12 and the first and second coefficients in c13, are in the significance state, for which the value is 1. All the remaining coefficients are coded in the cleanup pass and the columns c8, c9, c10 are coded by run-length with the unique single context that four contiguous coefficients in a column are all zero. Fig. 1*b* shows the column clustering result. The significant columns, (c1, c6, c7, c2), (c12, c13, c15, c8), are moved to the left with the shift record kept, while the other columns are only sequentially shifted afterwards. Significance state bits are increased from 4 to 10 in the significance propagation pass and the columns coded with the unique single context are also increased from 3 (i.e. c8, c9, c10) to 7 (i.e. c3, c4, c5, c9, c10, c11, c14) columns. Apparently, significant and insignificant coefficients are actually clustered by this rearranging technique.

In addition to the coded output of JPEG2000, the original positions of the rearranged columns must be recorded and transmitted as side information. The total overhead R consists of two parts: one bit is used to indicate whether a stripe is significant or not, and the number of bits needed to indicate the original location of a significant column. For example, if there are $N_{i,j}^{(k)}$ significant columns in a significant stripe that need to be rearranged in the kth bit-plane layer with the code block size $w \times w$, where w is chosen as an exact power of 2, then the total overhead R in bits can be calculated as

$$R = \sum_{k=0}^{P} \left(\frac{W}{4} \times S^{(k)} + \sum_{j=1}^{S^{(k)}} \sum_{i=1}^{m_j} (N_{i,j}^{(k)} \times \log_2 w) \right) \quad \text{bits} \qquad (3)$$

Where $w/4$ represents the number of stripes in the significant code block, $S^{(k)}$ is the number of significant code blocks in the kth bit-plane layer and m_j is the number of significant stripes in code block j. Finally, the side information is further compressed by arithmetic coding [5] to reduce the overhead.

Table 1: Bit rates (bpp) and bit-rate reductions (%) of reconstructed images by SBPC-JPEG2000 with same PSNR (dB) as by JPEG2000 of 'Lena', 'Barbara', 'Bike' and 'Woman'

		JPEG2000	SBPC-JPEG2000					
			$p=2$		$p=4$		$p=6$	
	PSNR	Rate (bpp)	Rate	Reduction	Rate	Reduction	Rate	Reduction
Lena	28.74	0.087	0.078	10.34%	0.082	5.75%	0.092	−5.75%
	32.57	0.250	0.245	2.00%	0.240	4.00%	0.245	2.00%
	35.21	0.495	0.490	1.01%	0.481	2.83%	0.485	2.02%
	38.00	0.993	0.983	1.01%	0.981	1.21%	0.996	−0.30%
Barbara	26.03	0.100	0.096	4.00%	0.098	2.00%	0.099	1.00%
	29.80	0.249	0.242	2.81%	0.240	3.61%	0.243	2.41%
	33.15	0.500	0.485	3.00%	0.482	3.60%	0.488	2.40%
	37.85	0.979	0.971	0.82%	0.964	1.53%	0.973	0.61%
Woman	33.64	0.100	0.093	7.00%	0.090	10.00%	0.095	5.00%
	35.74	0.250	0.245	2.00%	0.242	3.20%	0.243	2.80%
	37.90	0.498	0.490	1.61%	0.485	2.61%	0.487	2.21%
	39.09	0.995	0.980	1.51%	0.974	2.11%	0.984	1.11%
Bike	31.04	0.097	0.091	6.19%	0.090	7.22%	0.097	0.00%
	35.60	0.247	0.243	1.61%	0.241	2.43%	0.251	−1.62%
	38.12	0.499	0.490	1.80%	0.481	3.61%	0.500	−0.20%
	40.64	1.000	0.997	0.30%	0.991	0.90%	0.999	0.10%

Results: Daubechies 9-7 biorthogonal wavelet filters were utilised and the code block size was set as 64 × 64. 'Lena' and 'Barbara' images (size 512 × 512) with three decomposition levels as well as 'Bike' and 'Woman' (size 2560 × 2048) with five decomposition levels are used. Various bit-plane layers ($p = 2, 4, 6$) for determining the significance clustering were tested. Table 1 compares the bit rates of JPEG2000 VM 9.0 and SBPC-JPEG2000 with the same PSNR values. The bit-rate overhead of the side information is in the range 0.39–4.1%, which is included in the bit-rate calculation. For small p, e.g. rearranging based on the first two MSB planes, the improvement of SBPC-JPEG2000 over JPEG2000 is significant. However, when p is larger than 4, the bit-rate starts to increase because of the large overhead. For a large p, rate increasing, represented by a minus sign, is also possible. The optimal p depends on the rate. When $p = 2$, SBPC-JPEG2000 achieves a bit-rate reduction of 6.88% on average over all tested images at the rate of 0.1 bpp, while $p = 4$ achieves an average bit-rate reduction of 3.24% at medium bit-rate in the range 0.25–0.5 bpp. These results reveal that the proposed algorithm offers a more efficient way to encode the significant columns than JPEG2000.

Conclusion: The SBPC-JPEG2000 method is proposed, which uses a novel significant bit-plane clustering technique for image coding. The clustering overhead is carefully managed to be small while the coding efficiency is increased significantly. Furthermore, this clustering technique can be easily applied to many other image coding schemes, including video coding.

22 April 2004
Electronics Letters online no: 20045079
doi: 10.1049/el:20045079

T.T. Lu and P.C. Chang (*Department of Electrical Engineering, National Central University, Chung-Li, 320 Taiwan, Republic of China*)

References

1 ISO/IEC JTC1/SC29/WG1 N1855: 'JPEG 2000 Part I: Final Draft International Standard'. August 2000
2 ISO/IEC 10918-1 and ITU-T Recommendation T.81: 'Information Technology-JPEG-Digital Compression and Coding of Continuous-Tone Still Image-Part 1: Requirements and Guidelines'. 1994
3 Long, M., Tai, H.M., and Yang, S.: 'Quantisation step selection schemes in JPEG2000', *Electron. Lett.*, 2002, **38**, pp. 547–549
4 Lian, C.J., *et al.*: 'Analysis and architecture design of block-coding engine for EBCOT in JPEG2000', *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**, pp. 219–230
5 Witten, I.H., Neal, R.M., and Cleary, J.G.: 'Arithmetic coding for data compression', *Commun. ACM*, 1987, **30**, pp. 520–540