A New Bit-Plane Entropy Coder for Scalable Image Coding

Rong Zhang[1,2], Rongshan Yu[1], Qibin Sun[1], Wai-Choong Wong[1,2]
[1]Institute for Infocomm Research, A*STAR, Singapore, 119613
[2]Dept. of Electrical and Computer Engineering, National Univ. of Singapore, 117576
{stuzr, rsyu, qibin, lwong}@i2r.a-star.edu.sg

Abstract

Compression ratio and computational complexity are two major factors for a successful image coder. By exploring the Laplacian distribution of the wavelet coefficients, a new bit plane entropy coder is proposed in this paper. Compared with the state-of-the-art JPEG2000 entropy coder (EBCOT), the proposed coder achieves a 0.75% better lossless performance for 5 level 5/3 wavelet decomposition at block size 64×64 and 2.56% at block size 16×16. Experimental results also show PSNR improvements of about 0.13dB at 1bpp and 0.25dB at 2bpp on average for lossy compression. However, the gain in coding performance is not based on increasing computational complexity but instead a reduction by using a static arithmetic coder which avoids complicated adaptive procedure.

1. Introduction

In recent years, wavelet based embedded image coder is quite attractive in modern multimedia applications. Wavelet transform, bit plane coding and other techniques make embedded image coders practical, which not only provide efficient compression performance, distortion scalability, resolution scalability, but also other attractive features such as region of interest and random access.

Embedded zerotree wavelet coding (EZW) proposed by Shapiro [1], set partitioning in hierarchical trees (SPIHT) proposed by Said and Pearlman [2] and embedded block coding with optimal truncation (EBCOT) proposed by Taubman [3] are such successful examples of embedded image coders. EBCOT, with the best coding performance and other rich features, now serves as the entropy coder for the state-of-the-art image coding standard JPEG2000 [4][5].

EBCOT is an independently block coded, context based adaptive bit plane coder, which has two *Tiers*. In *Tier 1*, spatial correlation of the wavelet coefficients are finely modeled for each subband and each bit in the bit-plane is adaptively coded by one of the three fractional coding passes, significant propagation pass, magnitude refinement pass and clean up coding pass in a raster order. After all the coefficients have been compressed, in *Tier 2*, the Post Compression Rate Distortion Optimization algorithm (PCRD) is applied to organize each code block bitstream in an optimal way to minimize the distortion subject to the bit rate constraints and thus generate the output bitstream [4].

In scalable image coding, coding efficiency of the low-resolution blocks appear particularly important especially for low bit rate coding. However, in EBCOT, these blocks often have relatively fewer coefficients and a context adaptive coder (MQ coder) sometimes ends coding before it adapts to the local properties. Whilst, for these higher energy blocks, lower order bit planes are often uniformly distributed, wherein experimental results show that a context adaptive coder sometimes causes expansion rather than compression. Although EBCOT has the bypass mode called *lazy coding* which directly outputs raw bits of the lower bit planes, it has no systematic way to tell from which bit plane a *lazy coding* is more efficient. So, in spite of that EBCOT is quite a delicate context adaptive entropy coder, it is still inefficient in certain cases and such complex context adaptivity for some bit planes may seem unnecessary. Thus, regarding the fact that reduction of complexity often has more practical impact than a modest increase in compression ratio in some applications, it would be interesting to seek a coder which can achieve comparable compression ratio while reducing the complexity as much as possible.

In this paper, we present the Context-based Bit Plane Golomb Coder (CB-BPGC). Our experimental results show that instead of using the MQ coder, a static arithmetic coder combined with certain techniques is also quite promising on coding performance considering the lower complexity. The new algorithm explores coding based on the prior knowledge of wavelet coefficients distribution properties. It is well known that wavelet coefficients in the HL, LH and HH subbands are nearly Laplacian distributed [4]. Hence, by combining context modeling techniques with the embedded coding algorithm called Bit-Plane Golomb Coding (BPGC) [6], which is suitable for a Laplacian distributed source, a coder with modest coding efficiency is presented, without using the MQ coder but instead, a static arithmetic coder with much lower complexity.

This paper is organized as follows. The implementation detail of the proposed algorithm are introduced in Section 2;

in Section 3, we compare the coding performance and complexity of our algorithm with the EBCOT coder; conclusion remarks are then presented in Section 4.

2. Proposed CB-BPGC Entropy Coder

Bit-Plane Golomb Coding (BPGC), an embedded coding strategy for a source with Laplacian distribution, is first presented by R.Yu in [6]. It is now successfully implemented in the Advanced Audio Zip (AAZ) and the latest MPEG-4 Audio Scalable Lossless Coding (SLS) Standard [7]. We start this section with a brief review of that algorithm, followed by the details of implementing it together with the context modeling techniques for scalable image coding.

2.1. Bit-Plane Golomb Coding

Consider a Laplacian distributed source X, which has the probability density function given by,

$$f_X(x) = e^{-|x|\sqrt{2/\sigma^2}}/\sqrt{2\sigma^2} \tag{1}$$

where the magnitude of each sample $X_i(i=1,2...N)$ is binary represented by bit planes. If we constrain the source X with an independent and identically Laplacian distribution, the probability of the bit $b_{j,i}$ in bit plane B_j can be simply described as,

$$p_j = pr(b_{j,i} = 1) = 1 - (1 + \theta^{2^j})^{-1} \quad {\scriptstyle j=m,m-1...0} \tag{2}$$

where m is the most significant bit plane. The Laplacian distribution parameter θ can be estimated by the number of samples N and the absolute sum of the samples A: $\theta = e^{-\sqrt{2/\sigma^2}} \approx e^{-N/A}$. The approximate probability p_j for bit-plane B_j can be further obtained as follows:

$$Q_j^L = \begin{cases} 1/(1+2^{2^{j-L}}) & j \geq L \\ 1/2 & j < L \end{cases} \tag{3}$$

$$L = minL' \in Z|2^{L'+1}N \geq A \tag{4}$$

In that simple way, for a block of samples, with the calculated N and A, BPGC models the approximate probabilities of the bits in bit planes. The parameter L, which is called the lazy bit plane parameter (*lzbp*), indicates from which bit plane BPGC enters to the *lazy bit planes* where bits '0' and '1' are uniformly distributed, and also specifies the skew probability of bits in the *non-lazy bit planes* (the most significant bit plane m to the L bit plane) according to the bit plane distance *D2L:* $j - L$. When coded, bits in the *lazy bit planes* are directly outputted to the bitstream and the approximate probability of bits in the *non-lazy bit planes* can be easily obtained from (3) and then fed to the binary static arithmetic coder [6].

2.2. Context Modeling

In audio scalable coding, the good performance of BPGC is based on the constraint that the coding source is independent and identically distributed *(i.i.d.)* and the fact that the audio signal is a 1-D signal which has low correlation coding samples. However, image wavelet coefficients are heavily spatial correlated and that is why many image coders adopt an adaptive arithmetic coding procedure, while the static probability model used in BPGC would surely lose some efficiency. For a given bit plane, although we can get the approximate bit probability from the parameter *D2L*, the real probability of the bit is quite different, which is significantly affected by the neighborhood coefficients. For example, it is more likely for the current bit to be '1' when most of the nearby coefficients are '1'. But fortunately, BPGC can be easily combined with the image context modeling techniques, by which we can get probabilities of the bits different from each other according to their contexts.

Table 1: D2L Contexts

Ctxt.No.	0	1	2	3	4	5	6
D2L	≤ -3	-2	-1	0	1	2	≥ 3

D2L	3	2	1	0	-1	-2	-3,-4,-5...
m: 7 lzbp: 4	7	6	5	4	3	2	1,0
m: 9 lzbp: 6	9	8	7	6	5	4	3,2,1,0
m: 8 lzbp: 6		8	7	6	5	4	3,2,1,0

AC coded   Output Raw Bits

Figure 1: CB-BPGC bit plane coding example

Hence, the main contexts which affect the probabilities are *D2L* and neighborhood contexts *ctxt*. A detailed description of the 7 *D2L* contexts is listed in Table 1 and an example of bit plane coding relating to this *D2L* concept is given in Fig.1. *D2L* context 0 is for the *lazy bit planes*. For the neighborhood contexts, part of the contexts in EBCOT is adopted [4]. 9 contexts are modeled for coefficients which are about to be significant in the current bit plane and 3 contexts for the samples already significant in previous coded bit planes, while sign bits are outputted without any compression to reduce the coding complexity where 5 sign contexts are used in EBCOT. Codebooks for the probilities can be trained offline and pre-saved for different contexts.

2.3. Context-based BPGC Algorithm

By incorporating the image context modeling techniques, we extend the BPGC algorithm to Context-based BPGC (CB-BPGC) to loosen the *i.i.d.* constraint of BPGC. CB-BPGC uses the same Post Compression Rate Distortion Optimization algorithm in EBCOT *Tier 2* to pack the bitstream after the embedded coding is done. For the embedded coding part, Fig.2 illustrates the encoding process for the coefficient blocks in CB-BPGC.

As shown in Fig.2, after calculating the lazy bit plane parameter *lzbp*, some block classification is done in order to model the local coefficients properties in a better way. Observation shows that blocks with $L < 0$ (LOWE blocks), which are low entropy high frequency subband blocks, have quite a different *D2L* and *ctxt* related bits probabilities to

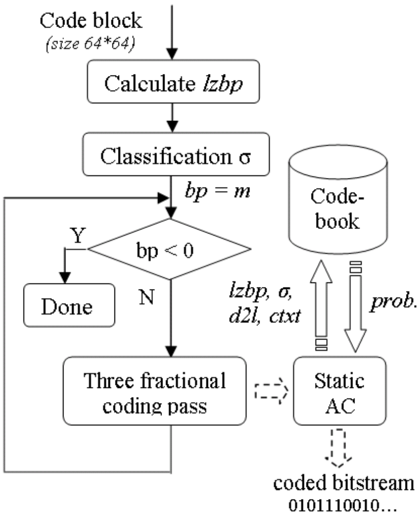


Figure 2: CB-BPGC encoding a block

those blocks with $L \geq 0$ (SIG blocks). In addition, for that SIG blocks, three different classes appear with distinct bits probabilities. Fig.3 gives an example. The three 64×64 blocks in the figure have the same $m = 6$ and $L = 3$, but the left one is smooth, the middle one seems more textual and the right one contains obvious edge. If each block is divided to smaller 8×8 sub-blocks (number in the sub-block area $subm_{x,y}$ indicates the most significant bit plane in the current sub-block), we can see that the smooth block has a smaller σ, the textual block has a median σ and the edge block has a larger σ, where σ is the standard deviation of $subm$ array ($\overline{subm}$ is the mean value),

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{x,y} (subm_{x,y} - \overline{subm})^2} \tag{5}$$

We can then trained the thresholds of the parameter σ to classify the three blocks. Blocks with the same class share the same codebook mentioned in last subsection.

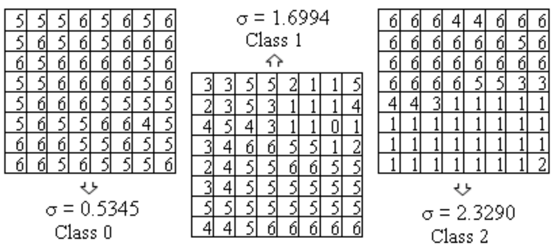


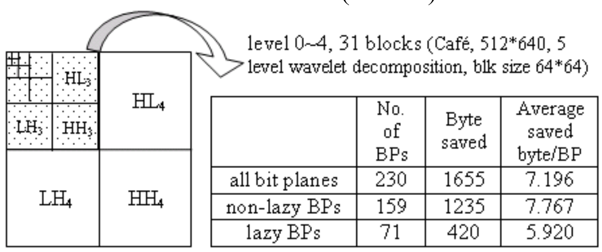
Figure 3: Block classification example: $m = 6, L = 3$ (block size: 64×64, divided into 8×8 sub blocks, numbers in the sub blocks are the most significant bit planes of the current sub blocks.)

After classification, CB-BPGC applies fractional bit plane coding passes, bitplane by bitplane, to get a finer embedded bitstream. A static binary arithmetic coder then compresses all the bits with the look-up probability from the codebook.

3. Experimental Results

The proposed coder is implemented with the well known Java implementation of JPEG2000 (*JJ2000*)[8]. A number of JPEG2000 grayscale images (10 typical JPEG2000 test images such as *cafe*, *fruits*, etc) are evaluated for both lossless and lossy performance.

Table 2 shows the results of lossless coding performance of 5 level wavelet decomposition at different types of block size for reversible 5/3 wavelet filter specified in the JPEG2000 starndard. The numbers of bit per pixel for each compressed image by the EBCOT coder and the proposed CB-BPGC are listed and the positive numbers in the *Perc.* columns indicate the percentage of CB-BPGC better than EBCOT while the negative ones are inverse. The average coding results show that CB-BPGC is more efficient in compression than EBCOT, especially for those images which are hard to compress, e.g. *baboon* and *cafe*, where an adaptive coding procedure probably fails to learn the complicated texture-like bit planes well. The compression results of the *non-lazy bit planes* for levels 0-4 coefficients of image *cafe* in Fig.4 give a detailed example for this. The *lazy bit planes* performance in Fig.4 also shows that it is more efficient to output the raw bits in the *lazy bit planes* instead of adaptive coding. In addition, EBCOT loses more efficiency in the case of smaller code block size, e.g. when code block size is 16×16, CB-BPGC is average 2.56% better for the 5/3 wavelet transform (Table 2).



	No. of BPs	Byte saved	Average saved byte/BP
all bit planes	230	1655	7.196
non-lazy BPs	159	1235	7.767
lazy BPs	71	420	5.920

Figure 4: Cafe example

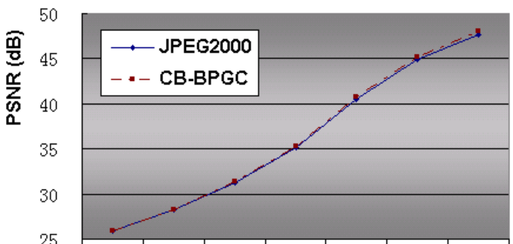
The average scalable compression performance of the images in Table 2 (Daubechies 9/7, 5 level decomposition, block size: 16×16) is listed in Fig.5, which shows that CB-BPGC outperforms EBCOT in terms of PSNR except at very low bit rates. The PSNR of CB-BPGC is more than 0.1dB for bitrate 1 bpp and about 0.25dB for bitrate 2bpp on average. But for the low bit rate, it is probably because the LL band coefficients is more like Rayleigh distribution where BPGC cannot model them well and another reason may be that we sacrifice some coding efficiency of the sign bit by direct transmission to reduce complexity where in low frequency subbands sign bits are compressible.

CB-BPGC is also lower complexity than JPEG2000. For the grey scale images lossless and lossy encoding, on average about 11.04% and 9.02% of the JPEG2000 encoding runtime are saved in CB-BPGC. By directly outputting the sign bits and bits in the *lazy bit planes* we reduce some bur-

Table 2: Lossless compression results for reversible DWT 5/3 LeGall (in bit/pixel)

Image	Size	64×64			32×32			16×16		
		J2K	BPGC	Perc.	J2K	BPGC	Perc.	J2K	BPGC	Perc.
baboon	500×480	6.166	6.020	2.36%	6.277	6.106	2.72%	6.626	6.412	3.22%
barb	720×576	6.249	6.143	1.69%	6.367	6.231	2.13%	6.728	6.553	2.61%
fruits	640×512	4.149	4.168	-0.46%	4.245	4.229	0.38%	4.538	4.451	1.91%
goldhill	720×576	4.645	4.609	0.78%	4.741	4.674	1.42%	5.058	4.937	2.39%
lena	512×512	4.620	4.568	1.12%	4.714	4.629	1.82%	5.022	4.871	3.01%
monarch	768×512	3.845	3.894	-1.28%	3.944	3.940	0.08%	4.237	4.150	2.04%
woman	512×640	4.238	4.234	0.10%	4.329	4.306	0.54%	4.619	4.532	1.89%
café	1024×1280	5.673	5.570	1.80%	5.791	5.671	2.07%	6.148	5.966	2.95%
tool	1280×1024	4.402	4.414	-0.28%	4.509	4.473	0.79%	4.826	4.708	2.44%
actors	1280×1024	5.408	5.320	1.62%	5.522	5.409	2.05%	5.873	5.690	3.12%
average		4.957	4.894	0.75%	5.062	4.967	1.40%	5.390	5.227	2.56%

den of the block context modeling in CB-BPGC. As it is shown in the Fig.4 *cafe* example, for the levels 0-4 code blocks, 30.9% of the bit planes are *lazy bit planes*($D2L \leq -3$), which can be transmitted directly. We can further reduce it by letting more bit planes be *lazy bit planes*, for example, bit planes with $D2L \leq -1$, where 57.8% of the bit planes in the Fig.4 *cafe* example can be directly outputted. The experiment also shows that the average lossless coding performance of the ten images is still better than EBCOT by 0.73% and 2.49% for block size 64×64 and 16×16 respectively. By sacrificing a little coding efficiency, complexity is reduced on these bit planes. In addition, the static arithmetic coder always has lower computational complexity than an adaptive arithmetic coders by avoiding the probability adaptive procedure. A recent research on computational complexity comparison of arithmetic coding algorithms [9] shows that with a present-day processor, encoding time of the static arithmetic coder is about 58.6% of that of JPEG2000 MQ coder for binary coding. Since a static arithmetic coder is used in CB-BPGC while EBCOT used the adaptive MQ coder, further reduction of complexity can be achieved.



	0.125	0.25	0.5	1	2	3	4
JPEG2000	25.918	28.253	31.250	35.184	40.513	44.951	47.690
CB-BPGC	25.904	28.278	31.320	35.306	40.758	45.244	48.092

Bitrate (bpp)

Figure 5: Average PSNR performance

4. Conclusion

We proposed a new entropy coder, CB-BPGC, for image scalable coding based on the statistical characteristics of the wavelet coefficients. By combining the embedded bit plane coder BPGC with the context modeling technique, CB-BPGC outperforms EBCOT on both lossless and lossy compression performance. Besides, computational complexity in CB-BPGC is also reduced compared to EBCOT.

References

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

[2] A.Said, W.Pearlman,"A new, fast and efficient image codec based on set partitioning in hierarchical trees" *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 6, pp. 243-250, Jun. 1996.

[3] D.Taubman,"High performance scalable image compression with EBCOT" *IEEE Trans. Image Processing*, Vol. 9, pp. 1158-1170, Jul. 2000.

[4] D.Taubman, and M.W.Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston/ Dordrecht/ London, 2002.

[5] ISO/IEC 15444-4:2000. Information technology-*JPEG2000 image coding system-Part 1: Core coding system*, 2000.

[6] R.Yu, C.C.Ko, S.Rahardja, X.Lin,"Bit-plane Golomb coding for sources with Laplacian distributions" *IEEE Int. Conf. Accoustics, Speech, and Signal Processing*, 2003.

[7] R.Yu, X.Lin, S.Rahardja, H.Huang,"Technical Description of I2R's Proposal for MPEG-4 Audio Scalable Lossless Coding (SLS): Advanced Audio Zip (AAZ)" ISO/IEC JTC1/SC29/WG11, M10035, Oct. 2003.

[8] http://jj2000.epfl.ch/

[9] A.Said, "Comparative analysis of arithmetic coding computational complexity" *HP Labs Tech. Reports*, HPL-2004-75, Apr.2004.