SUCCESSIVE BIT-PLANE RATE ALLOCATION TECHNIQUE FOR JPEG2000 IMAGE CODING

Y. M. Yeung, Oscar C. Au, Andy Chang
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Email: eeyym@ust.hk, eeau@ust.hk, eecax@ust.hk

ABSTRACT

A novel rate control scheme using successive bit-plane rate allocation (SBRA) is proposed for JPEG2000 image coding. By using the current rate-distortion information only, the proposed method can achieve a quality close to the post-compression rate-distortion (PCRD) optimization scheme adopted in JPEG2000. The proposed scheme can efficiently reduce both the computational cost and working memory size of the entropy coding process up to about 90%, in the case of 0.25bpp (1/32) compression. Without using the future rate-distortion information, the sequential property of the proposed method is highly suitable for real-time (or low delay) applications and implementation.

1. INTRODUCTION

With the increasing use of multimedia and network technologies, the efficient use of channel bandwidth is one of the main issues in transmitting image/video data. Rate control or rate allocation is necessary to control the bitrate of image/video coding such that it meets the channel bandwidth or end-to-end delay requirement. The final target of rate control is to allocate the target bitrate into an image/video such that the overall distortion can be minimized.

JPEG2000 [1] is a new international standard for still image coding. It can provide both objective and subjective image quality superior to existing standards. The encoding algorithm of JPEG2000 comprises discrete wavelet transform (DWT) and bit-plane MQ coder [2]. Unlike the conventional discrete cosine transform (DCT) based JPEG [3], the DWT based JPEG2000 has the intrinsic properties of signal-to-noise ratio (SNR) and resolution scalability. In JPEG, the rate is controlled by trying different values of quantization stepsizes (or quality factors). However, by using the bit-plane MQ coder, JPEG2000 can control the bitrate to meet the bitrate requirement precisely and easily.

The basic encoding algorithm of JPEG2000 is based on EBCOT (Embedded Block Coding with Optimized Truncation) [4]. The optimized truncation (rate-distortion optimization) is a rate control process that minimized the image distortion for a given bitrate. This process is applied after all the wavelet coefficients have been entropy encoded (compressed) and can be referred to as post-compression rate-distortion (PCRD) optimization.

By knowing the actual rate-distortion information of all compressed data, the PCRD technique attains to provide the minimum image distortion for a given bitrate. However, since it requires encoding all the data and storing all the encoded bitstream even though a large portion of the data needs not be sent out, most of the computation and memory usage could be redundant in this process. Also the PCRD is an off-line process such that the whole image needs to be completely encoded before sending out any data and hence long delay is possible.

In this paper, we propose a novel rate control scheme that can efficiently remove the computation and memory usage redundancy and provide a real-time (low delay) feature. Instead of using rate-distortion information of the whole image, the proposed scheme uses only the currently available rate-distortion information to control the bitrate and can obtain an image quality close to the optimal process. Without knowing the future rate-distortion information, the proposed causal scheme can be done in real-time such that part of the data can be sent out during the encoding process. This is very useful especially for the progressive transmission in JPEG2000 without long delay.

2. OVERVIEW OF JPEG2000

JPEG2000, as noted previously, is the new international standard for still image coding. In the following, we restrict the description to Part I of the standard that defines the core coding system. JPEG2000 is based on the discrete wavelet transform (DWT), scalar quantization, coefficient bit modeling, arithmetic coding and the post-compression rate control.

The DWT decomposes an image (or sub-image called tile) into LL, HL, LH and HH subbands as shown in Fig. 1. The subscripts of the subband symbols indicate the *resolution levels* that represent different resolution of the image/tile and LL_0 is the lowest resolution. The subbands consist of coefficients that represent the horizontal and vertical spatial frequency characteristics of the image/tile. Each subband is then quantized and divided into rectangular blocks (called *code blocks* in JPEG2000) with size typically 64x64. The quantized code block data is entropy coded using coefficient bit modeling and arithmetic coding. The coded data is organized by the post-compression rate control and finally outputted to the codestream in packets.

2.1. Entropy Coding (Tier-1 Coding)

The entropy coding process includes the coefficient bit modeling and arithmetic coding. This process is called *Tier-1* coding in JPEG2000. Tier-1 coding is essentially a bit-plane coding process and it is commonly used in wavelet based image coder [5-6]. In the Tier-1 coding process, code blocks are coded independently of one another. Each code block is coded from the most significant bit-plane (MSB) to the least significant bit-plane (LSB) by using three coding passes. The three coding passes are *significance pass*, *refinement pass* and *cleanup pass*. Each coefficient bit of a bit-plane is coded by one of the three coding passes. The coding passes will select the context of each coefficient bit using coefficient bit modeling. The pass data is then entropy coded by a context-based adaptive binary arithmetic coder (MQ coder).

2.2. Rate Control

Rate control in JPEG2000 is achieved by quantization and selection of the coding passes to be included in the codestream. The quantization process roughly controls the rate that is generally far from the target rate and is applied only once. The fine rate control is achieved by the selection of coding passes inclusion. Each coded code block has its bitstream data with a set of truncation (termination) points where each point is set at the end of each coding pass. The truncation points determine the number of coding passes inclusion in a code block.

The optimal selection process is achieved by the post-compression rate-distortion (PCRD) optimization as noted in section 1. This process had been described in [4] clearly. The optimization algorithm could be summarized as follow.

Let $\boldsymbol{D_i^{n_i}}$ and $\boldsymbol{R_i^{n_i}}$ be the distortion and rate at the truncation point $\boldsymbol{n_i}$ for code block $\boldsymbol{B_i}$ respectively. The optimal selection process is going to minimize the overall reconstructed image distortion $\boldsymbol{D}$ such that

$$\boldsymbol{D} = \sum_i \boldsymbol{D_i^{n_i}} \tag{1}$$

subject to the rate constraint

$$\boldsymbol{R} = \sum_i \boldsymbol{R_i^{n_i}} \leq \boldsymbol{R_{budget}} \tag{2}$$

where $\boldsymbol{R_{budget}}$ denotes the target bitrate.

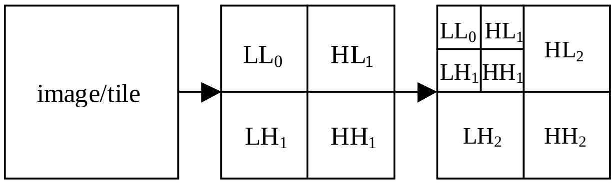

Fig. 1. Example of DWT decomposition into subbands.

At truncation point $\boldsymbol{n_i}$, the rate-distortion (R-D) slope is calculated by

$$\frac{\Delta \boldsymbol{D_i^{n_i}}}{\Delta \boldsymbol{R_i^{n_i}}} = \frac{\boldsymbol{D_i^{n_i-1}} - \boldsymbol{D_i^{n_i}}}{\boldsymbol{R_i^{n_i}} - \boldsymbol{R_i^{n_i-1}}} \tag{3}$$

Using the Lagrange multiplier technique [4,7], the optimization can be achieved by selecting the truncation points such that all included coding passes have the R-D slope greater than or equal to a constant value $\boldsymbol{\lambda}$ subject to the rate constraint (2). One should note here that the slopes must be strictly decreasing.

3. PROPOSED RATE CONTROL SCHEME

The optimal $\boldsymbol{\lambda}$, in section 2.2, can only be found by knowing the actual R-D slopes of all (past and future) truncation points. This requires Tier-1 coding of all the data and the whole encoded bitstream must be stored in memory even though a large portion of data will not be included in the final codestream after the rate control. A significant portion of computation power and working memory size is wasted on computing and storing the unused data. We call this portion of computation power and working memory size to be *redundant computation cost* and *redundant memory usage* respectively. Also the PCRD is an off-line process such that the entire image/tile needs to be completely encoded before sending out any data and hence long transmission delay is possible.

For a real-time rate control process, part of the data can be sent out during the encoding process. The only way to do this is to select the truncation points in parallel with the Tier-1 coding. This can also remove the redundant computation cost and redundant memory usage. However, without using the actual R-D slopes of the whole image/tile, it is impossible to find the optimal truncation points. So the problem now becomes finding the ‘good enough’ truncation points but not the optimal.

In section 2.2, the optimal selection is to find the truncation points such that the R-D slope

$$\frac{\Delta \boldsymbol{D_i^{n_i}}}{\Delta \boldsymbol{R_i^{n_i}}} \geq \boldsymbol{\lambda_{optimal}} \tag{4}$$

At truncation point n_i, the R-D slope (3) can only be computed after Tier-1 coding of that truncation point. We call n_i to be a future truncation point. In a real time rate control process, the future truncation points are unknown and hence cannot be used. Without using the future information, we find that the R-D ratio is a good approximation of the pattern of R-D slope as shown in Fig 2. The R-D ratio for code block B_i is defined as

$$\frac{D_i^j}{R_i^{j,n_i}} = \frac{\text{Distortion at bitplane } j}{\text{Rate for bitplane } j \text{ at truncation pt. } n_i} \tag{5}$$

where D_i^j is the distortion at bitplane j and R_i^{j,n_i} is the rate used for bitplane j at truncation point n_i. One should note that we are not going to approximate the value of R-D slope using R-D ratio. The R-D ratio is just an indicator showing the R-D relationship. In Fig 2, we see that both the R-D slope and the R-D ratio have similar pattern and thus R-D ratio could be a good indicator. Using the R-D ratio instead of R-D slope, the $\lambda_{optimal}$ in (4) cannot be used. So we approximate (4) by

$$\frac{D_i^j}{R_i^{j,n_i}} \geq \alpha_i^j \tag{6}$$

The α_i^j can be found by using the rate constraint such as (2). Let D_{remain} and R_{remain} be the remaining distortion and rate budget respectively such that

$$D_{remain} = D_i^j + \sum_{k \in B_{uncode}} D_k^0 \tag{7}$$

and

$$R_{remain} = R_{budget} - \sum_{k \in B_{coded}} R_k^{j,n_k} \tag{8}$$

where B_i is the current code block, B_{coded} and B_{uncode} are the coded and uncoded code block(s) respectively. Using the rate constraint

$$R_i^{j,n_i} + \sum_{k \in B_{uncode}} R_k^{j,n_k} \leq R_{remain} \tag{9}$$

and assuming α_i^j to be constant, (6) becomes

$$\frac{D_i^j}{R_i^{j,n_i}} \geq \frac{D_{remain}}{R_{remain}} \tag{10}$$

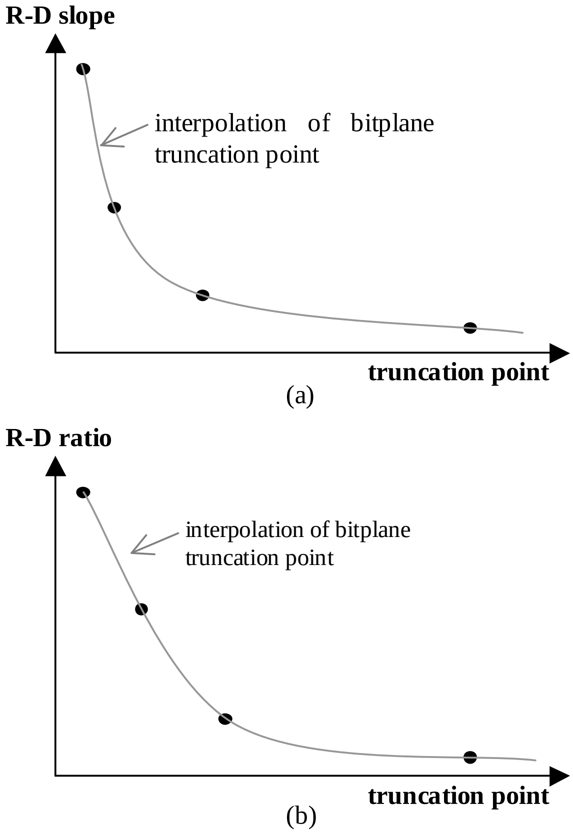

Fig. 2. Rate-distortion properties of (a) rate-distortion slope and (b) rate-distortion ratio.

Before actually doing the coding pass at truncation point n_i, the only unknown is R_i^{j,n_i} in (10). Then we can pre-compute a target rate for each bit-plane such that

$$Rt_i^j = D_i^j * \frac{R_{remain}}{D_{remain}} \tag{11}$$

Tier-1 coding of code block B_i will be stopped when

$$R_i^{j,n_i} \geq Rt_i^j \tag{12}$$

Using (11), the proposed rate control scheme can be simplified to the form as (12). Eqn. 12 is the core part of the proposed scheme and can be referred to as successive bit-plane rate allocation (SBRA).

We should note that we can stop the coding when Rt_i^j is small enough before the inclusion of coding pass. In other word, we can stop the coding when

$$Rt_i^j \leq T \tag{13}$$

In practical implementation, we set the threshold T as zero.

4. EXPERIMENTAL RESULTS

The proposed rate control scheme was tested on seven 512x512 images including "Barbara", "Boat", "Goldhill", "Lena", "Baboon", "Peppers" and "Zelda" as well as the three test images from the JPEG2000 test suite, "Bike", "Café" and "Woman". The proposed scheme was implemented in the reference software called "Jasper" [8] as defined in Part 5 of the standard. In all images, we used the Daubechies 9/7 bi-orthogonal wavelet filters with four level wavelet decomposition (or five resolution levels) and the code block size was 64x64.

Table 1 gives the results of complexity and memory reduction for two test images with different resolution. The computational cost is measured by a powerful profiling tool[1] in term of number of clock cycles used and the memory usage is measured in bytes. Compared with the rate control scheme in JPEG2000, the proposed scheme can efficiently reduce the computational cost and working memory size of Tier-1 coding where the Tier-1 coding can cost about 40% of total encoding time.

The proposed scheme degrades only 0.2dB and 0.4dB in 0.0625bpp and 0.25bpp respectively when compared with the optimal scheme in JPEG2000. Fig. 3 shows the average PSNR comparison results. As a reference, a similar fast rate control scheme that uses training model is mentioned in [9]. Its PSNR results show about 1dB loss from low to high bitrate. Using a different approach, our scheme can outperform the training model in [9].

5. CONCLUSION

In this paper, a novel rate control scheme is proposed for JPEG2000 image coding. The proposed causal scheme use current rate-distortion information only to control the rate. Without using any training model and the future rate-distortion information, this scheme can efficiently remove the redundant computation cost and redundant memory usage with PSNR quality close to the optimal scheme. This scheme also has the feature that part of the data can be sent out during the coding process. This feature is highly suitable for real time (or low delay) applications and implementation.

ACKNOWLEDGEMENT

This work was supported by Research Grants Council of Hong Kong SAR (HKUST6203/02E).

REFERENCES

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

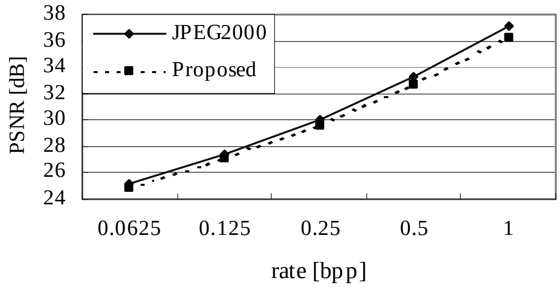

Fig. 3. Average PSNR results

[2] C. Christopoulos, et. al., "The JPEG2000 Still Image Coding System: An Overview," *IEEE Trans. on Consumer Electronics*, vol. 46, no. 4, Nov. 2000.

[3] G. K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Trans. Consumer Electronics*, vol. 38, no. 1, Feb. 1992.

[4] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, July 2000.

[5] J. M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[6] A. Said, 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 Tech.*, vol. 6, no. 3, pp. 243-250, June 1996.

[7] H. Everett, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," *Oper. Res.*, vol. 11, pp. 399-417, 1963.

[8] M. D. Adams, "JasPer project home page." http://www.ece.uvic.ca/~mdadams/jasper, 2000.

[9] T. Masuzaki, et. al., "JPEG2000 Adaptive Rate Control for Embedded Systems," *Proc. IEEE Int. Sym. on Circuits and Systems*, vol. 4, pp. 333-336, May 2002.

Table 1. Comparison results of complexity and memory usage. (1=JPEG2000, 2=proposed)

Lena(512x512)						
	Complexity			Memory		
Rate	(1)	(2)	Saved	(1)	(2)	Saved
0.125	$3.1\text{x}10^{8}$	$2.2\text{x}10^{7}$	93%	$9.1\text{x}10^{4}$	$4\text{x}10^{3}$	96%
0.25	$3.1\text{x}10^{8}$	$4.5\text{x}10^{7}$	85%	$9.1\text{x}10^{4}$	$8\text{x}10^{3}$	91%
0.5	$3.1\text{x}10^{8}$	$9.0\text{x}10^{7}$	71%	$9.1\text{x}10^{4}$	$1.6\text{x}10^{4}$	82%
Woman(2560x2048)						
	Complexity			Memory		
Rate	(1)	(2)	Saved	(1)	(2)	Saved
0.125	$7\text{x}10^{9}$	$4\text{x}10^{8}$	94%	$2\text{x}10^{6}$	$8\text{x}10^{4}$	96%
0.25	$7\text{x}10^{9}$	$8\text{x}10^{8}$	89%	$2\text{x}10^{6}$	$1.6\text{x}10^{5}$	92%
0.5	$7\text{x}10^{9}$	$1.6\text{x}10^{9}$	77%	$2\text{x}10^{6}$	$3.2\text{x}10^{5}$	84%

[1] Intel® Vtune™ Performance Analyzer 6.1