EMBEDDED IMAGE CODING WITH CONTEXT PARTITIONING AND QUANTIZATION

Hua Cai [1], *Xing San* [2*], *and Jiang Li* [1]

[1] Media Communication Group, Microsoft Research Asia, Beijing, China
[2] Dept. of Electronic Eng. & Information Science, University of Science and Technology of China

ABSTRACT

As a key part of universal source coding, context quantization is very important for improving compression performance. However, in most existing methods, the quantizer is trained offline and is fixed due to the complexity of finding a good quantizer and the significant overhead of representing the quantizer. This paper proposes a novel online context quantization approach that achieves high coding efficiency with low quantizer overhead and computational complexity. It first partitions the context into groups according to the number of significant context events. A layer-based context quantization is then applied on these groups. The proposed method is applied for embedded wavelet image coding. Compared with the JPEG2000 coder, up to 0.6 dB improvements can be achieved on the standard 512×512 test images. And more improvements are observed on images at lower resolutions.

1. INTRODUCTION

Entropy coding plays an important role in an image coding system. It achieves compression by coding the source symbols based on their statistical characteristics. In order to obtain good coding efficiency, the construction of a suitable model based on the source's characteristics is important. In recent years, higher order entropy coding based on universal source coding theory [1][2] has been attracting more and more research attention [3][4][5].

A key problem in universal sequential source coding of a discrete random sequence X_0, X_1, X_2... is the estimation of conditional probabilities $P(X_t|X^{t-1})$, where X^{t-1} denotes X_0, X_1, ..., X_{t-1}, the prefix of X_t. The estimation is based on a model. In practice, only a variable-order subset of X^{t-1}, called the context, C_t, is used to estimate $P(X_t|X^{t-1})$. However, it does not necessarily mean that the higher order modeling context leads to a shorter codelength. Since the conditional probabilities $P(X_t|C_t)$ have to be estimated from the corresponding symbol histograms in different conditioning states, if the order of the context model becomes too high, an image may not provide sufficient samples for the convergence of many symbol histogram to $P(X_t|C_t)$. Thus, high order context modeling can reduce coding efficiency. This problem is commonly known as *context dilution* or *model cost* [2]. To avoid this problem, the state-of-the-art image coding algorithm JPEG2000 [6] quantizes the context, C_t, into a relatively small number of conditioning states.

To deal with the context dilution issue more effectively, the optimal context quantization problem has been studied in-depth in [4][5]. Due to the complexity of finding good conditioning states and the significant overhead of representing the found states (or the quantizer), in these works, the conditioning states are trained offline from a training set and saved on the decoder side as well. However, as different image contents may have different statistical characteristics, the trained states may not always be efficient. Also, the impact of context dilution is different at different bit rates. For example, the problem may become more serious at a low bit rate since fewer samples can be provided, whereas it is not so terrible at a high bit rate. Thus, the fixed states may not be suitable for a wide range of bit rates. Furthermore, in addition to the above content adaptation and quantizer overhead issues, how to find a good context quantizer with acceptable complexity becomes an important problem, especially when a high order context model is used.

*The work presented in this paper was carried out in Microsoft Research Asia.

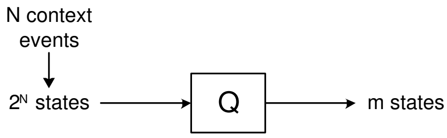

Figure 1: Context quantizer.

This paper proposes a novel online context quantization approach that achieves high coding efficiency with low quantizer overhead and computational complexity. It first partitions the context into groups according to the number of significant context events. A layer-based context quantization is then applied on these groups. The proposed approach is then applied for embedded wavelet image coding. Compared with the JPEG2000 coder, up to 0.6 dB improvements can be achieved on the standard 512×512 test images. And more improvements are observed on images at lower resolutions.

The rest of the paper is organized as follows. Section 2 presents the proposed context quantization approach, including performance analysis, context partitioning, and layer-based context quantization. Section 3 describes how to implement the proposed approach for embedded wavelet image coding. Experimental results are reported in Section 4 and conclusions are drawn in Section 5.

2. PROPOSED CONTEXT QUANTIZATION APPROACH

Consider a N-events context $c_1 c_2 ... c_N$, where $c_i = \{0, 1\}$ denotes the (binary) value of one of the previous samples. Without context quantization, the total number of conditioning states is 2^N, which would be quite large when many context events are considered. As shown in Fig. 1, the objective of a context quantizer is to merge these 2^N original states into a relatively small number of states. Now the problem is how to design a good quantizer, which should be of acceptable complexity and small overhead, such that the coding performance could be maximized.

2.1. Quantization Performance Analysis

The minimum bit rate required to code a random variable X under the condition of a given random variable C is the conditional entropy $H(X|C)$:

$$\begin{aligned} & H(X|C) \\ &= \sum_{s_i \in C} P(s_i) \sum_{x \in X} P(x|s_i) \log_2 \frac{1}{P(x|s_i)} \\ &= \sum_{s_i \in C} \sum_{x \in X} P(x, s_i) \log_2 \frac{1}{P(x|s_i)} \end{aligned} \tag{1}$$

where s_i denotes the unquantized conditioning state of C. If C is quantized to $M = Q(C)$, then the conditional entropy becomes $H(X|M)$:

$$\begin{aligned} & H(X|M) \\ &= \sum_{m_j \in M} P(m_j) \sum_{x \in X} P(x|m_j) \log_2 \frac{1}{P(x|m_j)} \\ &= \sum_{m_j \in M} \sum_{x \in X} \left[\sum_{s_i \in m_j} P(x, s_i) \right] \log_2 \frac{\sum_{s_i \in m_j} P(s_i)}{\sum_{s_i \in m_j} P(x, s_i)}. \end{aligned} \tag{2}$$

Let's then consider a very simple context quantizer in which only two conditioning states of C, say s_k and s_l, are quantized (merged) into a new conditioning state, while other states are not quantized. Thus the additional entropy introduced by this context quantization process is:

$$\begin{aligned} & \Delta H(X|M) \\ &= H(X|M) - H(X|C) \\ &= \sum_{x \in X} \left\{ [P(x, s_k) + P(x, s_l)] \times \log_2 \frac{P(s_k) + P(s_l)}{P(x, s_k) + P(x, s_l)} \right. \\ & \left. \quad - P(x, s_k) \times \log_2 \frac{P(s_k)}{P(x, s_k)} - P(x, s_l) \times \log_2 \frac{P(s_l)}{P(x, s_l)} \right\}. \end{aligned} \tag{3}$$

And it is clear that, if $P(x|s_k) = P(x|s_l)$, $\Delta H(X|M)$ will be zero. That is, there is no additional entropy introduced by the context quantization.

If $P(s_k)$ is fixed for different $P(x|s_k)$, then the derivative of $\Delta H(X|M)$ with respect to $P(x|s_k)$ is:

$$\begin{aligned} & \frac{\partial \Delta H(X|M)}{\partial P(x|s_k)} \\ &= \sum_{x \in X} \left\{ P(s_k) \times \log_2 \frac{[P(s_k) + P(s_l)] \times P(x|s_k)}{P(x|s_k) \times P(s_k) + P(x, s_l)} \right\} \\ &\Rightarrow \begin{cases} < 0, & \text{if} \quad P(x|s_k) < P(x|s_l) \\ = 0, & \text{if} \quad P(x|s_k) = P(x|s_l) \\ > 0, & \text{if} \quad P(x|s_k) > P(x|s_l) \end{cases} \end{aligned} \tag{4}$$

We then study the impact of different $P(s_k)$ and $P(s_l)$ for the given $P(x|s_k)$ and $P(x|s_l)$. Eqn. (3) can be rewritten as:

$$\begin{aligned} & \Delta H(X|M) \\ &= \sum_{x \in X} \left\{ P(s_k) P(x|s_k) \times \log_2 \frac{[P(s_k) + P(s_l)] \times P(x|s_k)}{P(s_k) P(x|s_k) + P(s_l) P(x|s_l)} \right. \\ & \left. \quad + P(s_l) P(x|s_l) \times \log_2 \frac{[P(s_k) + P(s_l)] \times P(x|s_l)}{P(s_k) P(x|s_k) + P(s_l) P(x|s_l)} \right\}. \end{aligned} \tag{5}$$

If $P(x|s_k)$ is fixed, then the derivative of $\Delta H(X|M)$ with respect to $P(s_k)$ is:

$$\begin{aligned} & \frac{\partial \Delta H(X|M)}{\partial P(s_k)} \\ &= \sum_{x \in X} \left\{ P(x|s_k) \times \log_2 \frac{P(s_k) + P(s_l)}{P(x, s_k) + P(x, s_l)} \right. \\ & \quad + \frac{1}{\ln 2} \times \left[\frac{P(x, s_k) + P(x, s_l)}{P(s_k) + P(s_l)} - P(x|s_k) \right] \\ & \left. \quad - P(x|s_k) \times \log_2 \frac{1}{P(x|s_k)} \right\} \\ &> 0, \quad \text{if} \quad P(x|s_k) \neq P(x|s_l) \end{aligned} \tag{6}$$

Therefore, from Eqn. (4) and Eqn. (6), two important criteria can be made for designing a good context quantizer that reduces the additional entropy. First, when several conditioning states s_i are merged into one state m_j, their conditional probabilities $P(x|s_i)$ should be as close as possible. Second, the probabilities $P(s_i)$ should be as small as possible.

2.2. Context Partitioning

Ideally, if $P(x|s_i)$ and $P(s_i)$ are known for any conditioning state s_i, an effective context quantizer can be found through the two criteria mentioned in Section 2.1. In practice, however, both of $P(x|s_i)$ and $P(s_i)$ are unknown on the decoder side. Thus the above two criteria cannot be utilized directly.

Let's consider the conditional probability $P(x|c_1 c_2)$ of two context events c_1 and c_2 (for simplicity, assume that c_1 and c_2 are independent of each other):

$$P(x|c_1 c_2) = \frac{P(x) P(c_1 c_2|x)}{P(c_1) P(c_2)} = \frac{P(x|c_1) P(x|c_2)}{P(x)}. \tag{7}$$

And for N independent context events $c_1 \sim c_N$, the conditional probability $P(x|c_1 c_2 ... c_N)$ equals:

$$P(x|c_1 c_2 ... c_N) = \frac{\prod_{i=1}^{N} P(x|c_i)}{[P(x)]^{N-1}}. \tag{8}$$

Let's assume that $\alpha = P(x = 1|c_i = 1)$ and $\beta = P(x = 1|c_i = 0)$ for all c_i, then:

$$P(x|c_1 c_2 ... c_N) = \frac{\alpha^n \beta^{N-n}}{[P(x)]^{N-1}} \tag{9}$$

where $n = \sum_{i=1}^{N} c_i$, the total number of significant context events (here c_i is significant means $c_i = 1$).

Usually, $P(x = 1|c_i = 1) \gg P(x = 1|c_j = 0)$ for any c_i and c_j. In other words, $\alpha \gg \beta$. Hence, it is clear from Eqn. (9) that conditioning states $s_i = c_1 c_2 ... c_N$ of different $\sum_{i=1}^{N} c_i$ will have quite different $P(x|s_i)$. Moreover, because $P(c_i = 0) \gg P(c_i = 1)$, $P(s_i)$ will decrease dramatically when $\sum_{i=1}^{N} c_i$ increases. Therefore, according to the number of significant events, $\sum_{i=1}^{N} c_i$, the 2^N conditioning states can be partitioned into $N + 1$ groups. Conditioning states of the same group are of the same $\sum_{i=1}^{N} c_i$. Statistically speaking, states in the same group have close values of $P(x|s_i)$, whereas for states in different groups their $P(x|s_i)$ values are quite different. Also, $P(s_i)$ decreases from the first to the last group.

After the partitioning, the original out-of-order conditioning states meet the above two criteria, and layer-based context quantization is ready to be applied.

2.3. Layer-Based Context Quantization

Now we have $N + 1$ groups of conditioning states, where the i^{th} group $\mathcal{S}_i$ contains C_N^i states. States in the same group are then organized into a layered structure: $\mathcal{S}_i = \{L_i(1), L_i(2), ...\}$. Each layer contains a number of quantized conditioning states. These layers are constructed in a stepwise manner from the bottom to the top: the last layer keeps the original conditioning states without any quantization; other layers, say $L_i(j)$, are generated by merging the neighboring two conditioning states in $L_i(j + 1)$; and in the top layer $L_i(1)$, all of the original states are quantized into one state. For example, if a group contains 14 conditioning states, then 5 layers can be created. The number of quantized states of each layer is 1, 2, 4, 7, and 14, respectively.

The context quantization efficiency of each layer can then be calculated as:

$$\lambda_i(j) = \frac{H(x|L_i(j)) - H(x|L_i(j+1))}{N(L_i(j+1)) - N(L_i(j))} \tag{10}$$

where $H(x|L_i(j))$ denotes the entropy under the given conditioning states $L_i(j)$; and $N(L_i(j))$ denotes the number of conditioning states in $L_i(j)$.

Thus the layer-based context quantization can be performed as follows: for a given slope λ, the most suitable layer $L_i(k)$ of the i^{th} group satisfies: $\lambda_i(k + 1) < \lambda \le \lambda_i(k)$. Of course, a preprocessing stage is required before searching to make $\lambda_i(j)$ decreasing when j increases. Moreover, for a given total number of quantized states, the most suitable layer of each group can also be found by solving a Lagrangian minimization problem. And some fast searching algorithms such as bisection can be used to reduce the search complexity.

Finally, it can be seen that the proposed layer-based context quantization does not require complicated computation, hence fast encoding is achievable. This is very important for a high order context model where many context events are considered. Also, the context partitioning method discussed in Section 2.2 ensures good quantization performance. On the other hand, only a small number of overhead bits are required to represent the found quantizer, which is important when encoding images of small resolution.

3. IMPLEMENTATION FOR EMBEDDED WAVELET IMAGE CODING

To show the effectiveness of our proposed context quantization approach, we apply it to an embedded wavelet image coding scheme. In this scheme, the original image is first transformed by a discrete wavelet transform in a *Mallat* decomposition structure [6].

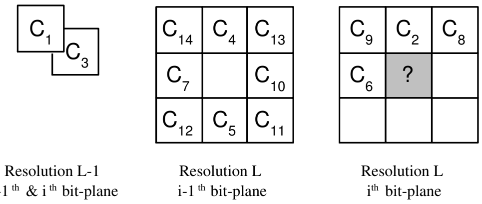

Figure 2: The 14 context events used in the zero coding pass.

The transformed coefficients are then separated into a series of bit-planes. Starting from the most significant bit-plane, three coding passes, i.e., *Zero Coding*, *Sign Coding*, and *Magnitude Refinement*, are applied for each bit-plane with a QM arithmetic coding engine.

1) *Zero Coding*: the zero coding pass encodes those coefficients which become significant at the current bit-plane. It is the most important pass among the three passes for an embedded wavelet encoder. Because of this, our proposed context quantization approach is used in this pass. We choose 14 context events for context quantization. Among them, 2 are from the lower resolution, 8 are from the same resolution but in the more significant bit-plane, and the last 4 are from the same resolution and the current bit-plane. As shown in Fig. 2, c_1 is the value of the corresponding pixel in the lower resolution and the more-significant bit-plane; c_3 is the value of the corresponding pixel in the lower resolution and the current bit-plane; c_4, c_5, c_7, and $c_{10} \sim c_{14}$ are the values of the eight neighboring pixels in the more significant bit-plane; and c_2, c_6, c_8, and c_9 are the value of the four neighboring pixels in the current bit-plane.

When a new bit-plane is encoded, the 2^{14} original conditioning states are first partitioned into 15 groups. States in each group are organized into different layers. Layer-based context quantization is then applied to generate the desired conditioning states with a fixed slope $\lambda = 3.5$. Since the number of samples increases as the bit-plane importance decreases, the number of quantized conditioning states also increases for a fixed λ.

2) *Sign Coding*: Once a coefficient is found significant in the zero coding pass, its sign value should be coded in this pass. Here we choose the same fixed context as that of JPEG2000 [6].

3) *Magnitude Refinement*: This pass refines the coefficients that are found significant in the more significant bit-plane(s). In this pass, we use only two conditioning states: a *one* state is set if the coefficient just changed to be significant in the nearest more significant bit-plane, otherwise the state is set to *zero*.

4. EXPERIMENTAL RESULTS

The standard test images, *Lena*, *Peppers*, and *Goldhill* (512×512, monochrome, and 8 bits/pixel) of different features are used for testing. To evaluate the performance under different image resolutions, these images are also down-scaled to 256×256 and 128×128 resolutions. The Daubechies 9/7-tap filters [7] are adopted for the wavelet transform. The images are decomposed into 7, 6, and 5 levels for the 512×512, 256×256, and 128×128 resolutions, respectively.

Table 1 lists the results of our new image codec with a comparison to JPEG2000 under the same file size. In our comparison, the JPEG2000 codec [8] generates a one layer bitstream with a 64×64 code-block size. Also note that we do not apply rate-distortion optimization in our scheme whereas it is used by JPEG2000 to further improve the performance. For the original 512×512 resolution, up to 0.6 dB improvement is achieved compared with JPEG2000. And more improvements can be observed for images at lower resolutions. This is mainly because our proposed method provides content-dependent and rate-adaptive conditioning states for different images and different bit rates, whereas in JPEG2000 only fixed conditioning states are used. Moreover, since the spatial correlation of wavelet coefficients becomes less efficient in images of low resolutions, the coding efficiency of JPEG2000 drops significantly when the image resolution reduces.

The encoding speed of this new image codec is also tested on a PC of a 2.8GHz Pentium CPU with 512M RAM. Although there is no source code optimization in our codec, we can still achieve an acceptable encoding speed (shown in Table 1). We believe that, by proper source code optimization, faster encoding is achievable.

5. CONCLUSIONS AND DISCUSSIONS

As a key part of universal source coding, context quantization is very important for improving compression performance. In this paper, we propose a novel online context quantization approach that achieves high coding efficiency with low quantizer overhead and computational complexity. This approach is then implemented for embedded wavelet image coding. For the original 512×512 image resolution, up to 0.6 dB improvement is achieved compared with JPEG2000. And more improvements can be observed for images at lower resolutions.

Since the proposed approach provides an effective way to handle a high order context model, it can be easily applied for color image coding, where context events may come from different color components. This is one of our directions for future works.

6. REFERENCES

[1] J. Rissanen, "A universal data compression system," *IEEE Transactions on Information Theory*, vol. 29, pp. 656–664, Sept. 1983.

[2] J. Rissanen, "Universal coding, information, prediction and estimation," *IEEE Transactions on Information Theory*, vol. 30, pp. 629–636, July 1984.

[3] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Transactions on Image Processing*, vol. 6, pp. 656–664, May 1997.

[4] J. Chen, "Context modeling based on context quantization with application in wavelet image coding," *IEEE Transactions on Image Processing*, vol. 13, pp. 26–32, Jan. 2004.

[5] S. Forchhammer, X. Wu, and J. D. Andersen, "Optimal context quantization in lossless compression of image data sequences," *IEEE Transactions on Image Processing*, vol. 13, pp. 509–517, Apr. 2004.

[6] David Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[7] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, Apr. 1992.

[8] ISO/IEC JTC1/SC29/WG1, "JPEG2000 Verification Model 7.2," May 2000.

Table 1: Performance evaluation of the new image codec. The bitstream is truncated at four rates, from bit-plane 32 to 4. In the table, *'Length'* denotes the total number of bytes after truncation (including all of the overhead), and *'States'* denotes the number of quantized conditioning states of the truncated bit-plane.

512×512

Image	Length / States	Time (sec)	PSNR (dB)	JPEG2000 PSNR (dB)
Lena	2459 / 54	0.312	29.25	28.62
	5324 / 58	0.583	32.48	32.10
	10941 / 62	0.792	35.67	35.38
	22763 / 114	1.104	38.87	38.69
Peppers	2325 / 82	0.364	29.56	29.04
	4829 / 178	0.632	32.49	32.12
	9923 / 191	0.887	35.12	34.85
	24533 / 199	1.151	37.93	37.72
Goldhill	2515 / 47	0.344	27.40	27.20
	6835 / 103	0.629	30.16	30.03
	17140 / 210	0.957	33.56	33.44
	38539 / 238	1.187	37.80	37.68

256×256

Image	Length / States	Time (sec)	PSNR (dB)	JPEG2000 PSNR (dB)
Lena	1130 / 58	0.153	26.90	26.31
	2642 / 62	0.252	30.51	30.01
	5239 / 63	0.302	34.61	34.07
	9412 / 67	0.386	38.96	38.51
Peppers	1077 / 36	0.104	27.31	26.60
	2363 / 39	0.201	31.15	30.35
	4590 / 101	0.261	34.94	34.29
	8453 / 114	0.353	38.78	38.37
Goldhill	935 / 35	0.102	25.80	25.42
	2653 / 49	0.169	28.91	28.56
	6547 / 112	0.338	32.79	32.53
	13303 / 140	0.422	37.69	37.47

128×128

Image	Length / States	Time (sec)	PSNR (dB)	JPEG2000 PSNR (dB)
Lena	527 / 17	0.044	24.51	23.48
	1261 / 47	0.061	28.66	27.60
	2339 / 49	0.073	33.35	32.52
	3845 / 49	0.102	38.38	37.42
Peppers	510 / 16	0.039	24.79	23.52
	1155 / 52	0.055	28.94	27.78
	2175 / 55	0.071	33.33	32.20
	3695 / 56	0.102	38.07	37.14
Goldhill	381 / 12	0.037	24.05	22.97
	1138 / 56	0.048	27.50	26.35
	2593 / 67	0.073	31.95	31.21
	4699 / 69	0.117	37.59	37.01