CONTEXT MODELING AND ENTROPY CODING OF WAVELET COEFFICIENTS FOR IMAGE COMPRESSION

Xiaolin Wu[1] *Jian-hua Chen*[2]

[1] Department of Computer Science
University of Western Ontario
London, Ontario, Canada N6A 5B7
[2] Department of Information Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong

ABSTRACT

In this paper we study the problem of context modeling and entropy coding of the symbol streams generated by the well-known EZW image coder (embedded image coding using zerotrees of wavelet coefficients). We present some simple context modeling techniques that can squeeze out more statistical redundancy in the wavelet coefficients of EZW-type image coders and hence lead to improved coding efficiency.

1. INTRODUCTION

There seems to be a general consensus that wavelet image coders, e.g., [5, 4], are so far the best performed lossy compression techniques in the range of medium to low bit rates. The success of wavelet image coders is largely due to the good local properties of wavelet transforms in both frequency and spatial domains. That is, with wavelet transforms, one can identify both the spectral and spatial locations of significant image features. This localization feature of wavelet transforms is extremely desirable for image compression, because one can then economically allocate bits to frequency ranges and spatial locations that are important to visual quality of reconstructed images. Good examples of utilizing wavelet localization property for image compression are the well-known technique of embedded image coding using zerotrees of wavelet coefficients (EZW algorithm) [5], and its variants such as [4, 6]. The EZW algorithm is in essence an integrated method of quantizing and entropy-coding wavelet coefficients. The quantizer employed by the zerotree technique is one of successive scalar quantization resembling bit plane encoding. Despite using scalar, successive quantization which is theoretical inferior to one-pass vector quantizer, image coders of the EZW type obtained some of the best compression results on common test images that were reported in the literature, while facilitating progressive reconstruction and transmission.

The good performance of zerotree-based wavelet coders is due to two important properties of the zerotree data structure. First, zerotrees, by taking advantage of the localization property of wavelet transform and the fact that natural images tend to have decaying spectrums, organize small wavelet coefficients into a quadtree hierarchy for compact encoding. A quadtree of insignificant wavelet coefficients (zeros after quantization), called zerotree in EZW, can be considered as a two-dimensional zero run. Embedding a zerotree structure into a wavelet transform for entropy coding of the wavelet coefficients has the same effect as zig-zag scanning and run-length encoding of DCT coefficients.

Second, the successive quantization scheme used by the EZW algorithm produces very small alphabets to facilitate adaptive entropy coding. There are only 2, 3 or 4 distinct symbols in particular segments of the zerotree code stream. From universal source coding point of view, a source drawn from a smaller alphabet can be better modeled as a Markov chain, and hence more efficiently coded than a source from a larger alphabet. Since the number of states of a Markov model increases exponentially in alphabet size, adaptive entropy coding based on Markov modeling quickly becomes impractical and inefficient as the number of source symbols increases. The zerotree technique cleverly reduces the alphabet of wavelet coefficients to no more than four symbols by bit plane encoding. This alphabet reduction technique shares the same spirit as binary arithmetic coders that encode an m-ary symbol as outcomes of a sequence of binary decisions [1].

In this paper we will reexamine the problem of entropy coding of quantized wavelet coefficients in the framework of zerotrees. Specifically, we propose some more sophisticated context modeling techniques than the second-order Markov modeling suggested by the original zerotree paper [5]. In his widely-cited work [5] Shapiro observed "the coding gains provided by using this simple Markov conditioning may not justify the added complexity and using a single histogram strategy for the dominant pass performs almost as well (0.12 dB worse for Lena at 0.25 bpp)." This research showed, however, that there exists much higher statistical redundancy than what might be implied by Shapiro's observation in the symbol stream of the so-called dominant pass of the EZW algorithm.

2. CONTEXT MODELING OF EZW COEFFICIENTS AND MODEL COST

Shapiro's negative observation is true in the sense that simply increasing the order of Markov modeling will get diminishing compression gains. But this does not imply that higher-order dependency between wavelet coefficients does not exist. Instead, it points to a pivotal issue called *model cost* in universal source coding [2]. Intuitively speaking, the better the model fits the source, the shorter the codelength an entropy coder like adaptive arithmetic coder can achieve. But in universal source coding in which no prior knowledge of the source is available, the model itself must be either explicitly sent to the decoder or learned on the fly from the symbol stream. In the former case, we need to add side information to the total description length of the source. In the latter case, the learning takes time to fit a statistical model to the source. The higher the model complexity (i.e., the more model parameters), the longer the time to "learn" the model. Before the model converges to the underlying statistics through on-line learning, entropy coding cannot reach the minimum codelength of the coded symbols. Thus not only how well the model fits the source and also how fast the fit can be achieved affect the total description length. Since in either case the model complexity contributes to the total description length of the source, Rissanen analytically formulated this phenomenon as the so-called "model cost" in his theory of minimum description length.

Entropy coding of quantized EZW coefficients is just a special form of lossless image coding with symbol dependency of structures unique to zerotrees. The efficiency of any lossless image codec is governed by how well it can model the high-order statistical redundancy at a low model cost. Recently we made progress in context modeling for lossless image coding, and developed CALIC (Context-based, Adaptive, Lossless Image Codec [7, 8]) which is the current state of the art. In the reported work we generalized some of CALIC's context modeling techniques to wavelet coefficient coding, and achieved better rate-distortion performance than existing methods.

Due to the popularity of the EZW algorithm in this research community, we simply adopt the same terminology of Shapiro's original paper [5] without redescribing the algorithm. The EZW symbols are drawn from three different alphabets $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{P, N, ZR, IZ\}$, and $\Sigma_3 = \{P, N, Z\}$. Letters 0 and 1 in Σ_1 are the outputs of the EZW algorithm when it is in a subordinate pass to refine quantization precision, being 0 if a significant coefficient is below the current quantizer threshold or 1 otherwise. Letters P, N, ZR, and IZ of Σ_2 stand for positive significant coefficient, negative significant coefficient, zerotree root, and isolated zero, the four cases in a dominant pass of the EZW algorithm over subbands other than the three highest resolution subbands HH, HL, and LH. Letters P, N, and Z of Σ_3 stand for the outputs of a dominant pass over subbands HH, HL, and LH in which there is no need to distinguish between zerotree roots and isolated zeros. We observed that bit streams out of Σ_1 in subordinate passes were hardly compressible even by high-order entropy coding. Some researchers independently had the same observation [4]. Thus we simply output the bit streams of Σ_1 as they are, and focus on context modeling and entropy coding of symbol streams out of Σ_2 and Σ_3.

Suppose that we are to sequentially encode a sequence of EZW symbols $x_1, x_2, \cdots, x_n$. The symbol sequence is determined by the way in which an EZW-type algorithm scans the EZW coefficients. The minimum codelength of the sequence in bits is given by

$$-\log_2 \prod_{i=1}^{n} P(x_i | x_{i-1}, \cdots, x_1). \qquad (1)$$

Arithmetic coding can approach this optimal codelength [3]. But $P(x_i|x_{i-1}, \cdots, x_1)$, $1 < i \leq n$, is generally unknown in practice, and has to be estimated on the fly based on past observations in the coding process. In universal source coding literature a mechanism of estimating $P(x_i|x_{i-1}, \cdots, x_1)$ is often called a statistical model of the source. The set of past observations on which the probability of the current symbol is conditioned is called modeling context. Clearly, it is the model that determines the rate at which we can encode the symbol sequence. But (1) does not necessarily mean that higher-order modeling context leads to shorter codelength. The number of possible conditioning states grows exponentially with the order of the context. Since conditional probabilities $P(x_i|x_{i+1}, \cdots, x_1)$ have to be estimated on the fly by corresponding symbol histograms in different conditioning states, an image may not provide sufficient samples for the convergence of too many symbol histograms to $P(x_i|x_{i-1}, \cdots, x_1)$. In other words, too large a modeling context spreads counting statistics too thin among all possible modeling states to reach good conditional probability estimates. The codelength will actually increase when the order of modeling contexts gets too high despite the fact that conditional entropy is monotonically non-increasing in the order of context modeling, i.e., $H(x_i|x_{i-1}, \cdots, x_{i-k-1}) \leq H(x_i|x_{i-1}, \cdots, x_{i-k})$. Thus high order of context modeling is more than a problem of high time and space complexities, it can reduce coding efficiency as well. This problem is commonly known as "context dilution" and formulated by Rissanen analytically as so-called "model cost" [2]. Intuitively speaking, the better the model fits the source, the shorter the codelength an adaptive entropy coder can achieve. But in universal source coding in which no prior knowledge of the source is available, the model itself must be either explicitly sent to the decoder or "learned" on the fly from the symbol stream. In the former case, we need to add side information to the total description length of the source. In the latter case, the learning takes time to fit a statistical model to the source. The higher the model complexity (i.e., the more model parameters), the longer the time to "learn" the model. Before the model converges to the underlying statistics through on-line learning, entropy coding cannot reach the minimum codelength of (1). In either case, the context model has a cost to the total description length. The pivotal issue of this research is to find modeling contexts that capture statistical dependency between EZW symbols from Σ_2 and Σ_3 but at a small model cost.

3. SYMBOL BINARIZATION FOR ENTROPY CODING

The EZW symbols can take on four or three possible letters depending on if they are drawn from Σ_2 or Σ_3. But letters P and N are equally probable because most wavelet coders are based on least-squares approximation criterion. Having this preknowledge, context modeling can reveal no unknown structures of the source by distinguishing P and N to the benefit of compression. On the contrary, it leads to poorer coding efficiency by increasing the model cost (the number of conditioning states) for nothing in return. Thus we combine P and N into one case called significant (denoted by letter S) to reduce the model cost and improve coding efficiency. In this work we reduce original alphabets $\Sigma_2 = \{ZR, IZ, P, N\}$ and $\Sigma_3 = \{Z, P, N\}$ to a binary alphabet $\Sigma = \{Z, S\}$. The use of binary alphabet Σ in both cases of Σ_2 and Σ_3 facilitates binary adaptive arithmetic coding which is computationally very efficient. If the current EZW symbol x is S (either P or N), one more bit is used to tell apart P and N. It is pointless to entropy code this bit because P and N are equally probable. Similarly, if $x = Z$ and $x \in \Sigma_2$ another bit is required to distinguish between ZR and IZ. In the latter case, we do entropy code this additional bit because the probabilities of ZR and IZ can differ significantly in different modeling states.

Given an EZW symbol x, consider a modeling context that involves seven EZW symbols related to x in the spatial domain: p, w, n, nw, ne, ww, and nn, where p is the parent of x, w, n, nw, ne, ww denote the EZW symbols located to the west, north, northwest, northeast, west-west, and north-north of x. Note that p, w, n, nw, ne, ww, and nn are processed prior to x in the scan order given by [5], hence they are available to both encoder and decoder. Clearly, to utilize the symbol dependency, we should restrict w, n, nw, ne, ww, and nn to be in the same subband as x. If x is at or near the boundary of its band, we simply duplicate the boundary symbols. The three most related symbols to x are p, n and w. First, we set three modeling bits b_p, b_w, and b_n to 1 or 0 depending on whether p is significant, whether w just becomes significant in the current dominant pass, and whether n just becomes significant in the current dominant pass. Based on the three modeling bits the context modeler estimates eight conditional probabilities $P(x|b_pb_wb_n)$ for $b_pb_wb_n = 000, \cdots, 111$. In our experiments, adaptive binary arithmetic coder that uses the estimated $P(x|b_pb_wb_n)$ achieved slightly shorter codelength than the original Shapiro's result on test image "lena" [5]. In order to exploit higher order symbol dependency in adaptive entropy coding, we consider more neighbors of x in addition to w and n. Apparent candidates for additional modeling events are nw, ne, ww, and nn. An attempting next step toward higher compression is to increase the order of context modeling to include all these modeling events. But in our experiments, the use of an additional modeling event reduced the codelength by a very small amount. The inclusion of more than four modeling events into the model context actually increased the codelength. This observation manifested the negative impact of model cost on the codelength. Without care the model cost starts to cancel possible compression gains brought by context modeling even when the order of the modeling exceeds three. In order to turn higher-order statistical redundancy into real compression gains we have to find ways of reducing the model cost.

4. QUANTIZATION OF MODELING EVENTS

A useful technique to reduce model cost is quantization of modeling events [7]. In fact, we have already effected a binary quantization of modeling events when forming b_w and b_n above. This can be easily seen by noting that b_w and b_n are set based on if the EZW coefficients at locations of w and n exceed the current but below the previous quantization threshold in successive dominant passes of the EZW algorithm, and by noting that the quantization threshold is lowered by half in the successive dominant passes. The distribution of random variable x depends on the continuous magnitudes of its neighboring EZW coefficients. But we have to quantize the EZW coefficients when forming the model contexts otherwise the model cost (the number of conditioning states) would be far too high. Likewise, it is necessary to quantize other EZW coefficients involved in context modeling. We set b_{nw}, b_{ne}, b_{ww} and b_{nn} to 1 if the EZW coefficients at the corresponding locations are significant up to the current dominant pass, otherwise to 0. Furthermore, we set b'_w and b'_n to 1 if the EZW coefficients at w and n were found to be significant in a previous dominant pass. The modeling events b'_w and b'_n together with b_w and b_n can capture statistical redundancy between successive dominant passes of the EZW algorithm. Note that b'_w and b_w in combination will provide a finer quantization of the EZW coefficient at w, and so will b'_n and b_n.

In the above we have created six more binary modeling events b_{nw}, b_{ne}, b_{ww}, b_{nn}, b'_w and b'_n. Without further context quantization, adding those binary modeling events will increase the number of conditioning states by $2^6 = 64$ times. We need to reduce this number drastically. To this end, we combine the six binary modeling events into two b_1 and b_2: $b_1 = 0$ if $b'_w + b'_n + b_{nw} + b_{ww} = 0$ and $b_1 = 1$ otherwise; $b_2 = 0$ if $b'_w + b'_n + b_{ne} + b_{nn} = 0$ and $b_2 = 1$ otherwise. This represents a vector quantization of six-dimensional binary random vectors into four codewords. Even with this rather aggressive context quantization scheme, we still found that adaptive entropy coding by estimating $P(x|b_pb_wb_nb_1b_2)$ on the fly gave slightly longer codelength than by estimating $P(x|b_pb_wb_nb_1)$ or $P(x|b_pb_wb_nb_2)$, whereas adaptive entropy coding driven by estimated $P(x|b_pb_wb_nb_1)$ or $P(x|b_pb_wb_nb_2)$ outperforms that by estimated $P(x|b_pb_wb_n)$. The dropping coding efficiency from four to binary five modeling events indicated a critical point at which the model cost gets high enough to render higher order context modeling counterproductive.

5. WEIGHTED PROBABILITY ESTIMATION IN MULTIPLE CONTEXTS

Since adaptive entropy coding benefits from estimated $P(x|b_pb_wb_nb_1)$ or $P(x|b_pb_wb_nb_2)$ but not from $P(x|b_pb_wb_nb_1b_2)$, an interesting question is if we can make use of both b_1 and b_2 together with b_p, b_w, and b_n without increasing the model cost and get a shorter codelength? The answer is yes. Instead of forming a modeling context by the Cartesian product of five binary modeling events, we create two modeling contexts: $C_1 = b_pb_wb_nb_1$ and $C_2 = b_pb_wb_nb_2$. Let $p(x|c_1)$ and $p(x|c_2)$ be estimated conditional probabilities $p_{X|C_1}(x|c_1)$ and $p_{X|C_2}(x|c_2)$ at any given moment of adaptive entropy coding, and $l(c_1)$ and $l(c_2)$ be the average codelengths of past symbols in the modeling contexts c_1 and c_2. We assign to the current EZW symbol x the weighted probability

$$p(x|c_1 \cup c_2) = \frac{2^{l(c_1)}p(x|c_1) + 2^{l(c_2)}p(x|c_2)}{2^{l(c_1)} + 2^{l(c_2)}} \qquad (2)$$

in adaptive arithmetic coding of x. Since $p(x|c_1)$ and $p(x|c_2)$ are probability measures on x given c_1 and c_2, $p(x|c_1 \cup c_2)$

Algorithms	Bit Rates			
	0.125	0.15	0.25	0.5
This Work	31.16	31.96	34.25	37.19
Shapiro (EZW)	30.23		33.17	36.28
Said,Pearlman	31.1	31.9	34.1	37.2

Table 1. Coding results for 'lena'

that is a weighted sum of $p(x|c_1)$ and $p(x|c_2)$ is also a probability measure on x. Our experiments showed that adaptive arithmetic coding of x based on $p(x|c_1 \cup c_2)$ improved the coding efficiency by about 3% over that on $p(x|c_1)$ or on $p(x|c_2)$ alone. The key idea here is to let $p(x|c_1 \cup c_2)$ have contributions of both modeling events b_1 and b_2 but without increasing the model cost. We only compute $p(x|c_1)$ and $p(x|c_2)$ in modeling contexts of four rather than five binary events. Probability estimates $p(x|c_1)$ and $p(x|c_2)$ are sample histograms, and each past observation is used to update both $p(x|c_1)$ and $p(x|c_2)$. In this way each sample is counted twice in computing $p(x|c_1 \cup c_2)$, whereas it could only be counted once in estimating $P(x|b_p b_w b_n b_1 b_2)$. This technique effectively increases the number of samples used by probability estimation, thus it alleviates the problem of context dilution which is caused by inclusion of both b_1 and b_2 in context modeling.

6. PERFORMANCE COMPARISON

The PSNR numbers for the proposed adaptive entropy coding method coupled with the EZW algorithm at various bit rates on test image "lena" are listed in Table 1. For comparison purposes, we also include in the table the performance figures on the same test image of the EZW algorithm and the recent Said and Pearlman's method (its arithmetic coding version). Our method significantly outperformed the original EZW algorithm at all bit rates, and it fared slightly better than Said and Pearlman's method for bit rates 0.25 and below. The latter is generally considered as the current performance leader.

REFERENCES

[1] C. B. Jones, "An efficient coding system for long source sequences", *IEEE Trans. Info. Theory*, vol. 27, pp. 280-291, 1981.

[2] J. Rissanen, "Universal coding, information, prediction, and estimation", *IEEE Trans. Info. Theory*, vol. 30, pp. 629-636, July 1984.

[3] J. Rissanen and G. Langdon, "Arithmetic coding", *IBM J. Res. and Devl.*, vol. 23, no. 2, pp. 149-162, 1979.

[4] A. Said and W. A. Pearlman, "New, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE Trans. Circ. & Sys. Video Tech.*, vol. 6, no. 3, pp. 243-249, June 1996.

[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] C. Y. Teng and D. L. Neuhoff, *"Quadtree-Guided Wavelet Image Coding"*, DCC'96, pp. 406-415.

[7] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling", *IEEE Trans. Image Processing*, (to appear).

[8] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec", *IEEE Trans. on Communications*, (to appear).