CALIC - A CONTEXT BASED ADAPTIVE LOSSLESS IMAGE CODEC

Xiaolin Wu[1] *Nasir Memon*[2]

[1]Department of Computer Science, The University of Western Ontario, London, Ontario, Canada N6A 5B7F
[2] Computer Science Department, Northern Illinois University, Dekalb, IL 60115

ABSTRACT

We propose a context-based, adaptive, lossless image codec (CALIC). CALIC obtains higher lossless compression of continuous-tone images than other techniques reported in the literature. This high coding efficiency is accomplished with relatively low time and space complexities. CALIC puts heavy emphasis on image data modeling. A unique feature of CALIC is the use of a large number of modeling contexts to condition a non-linear predictor and make it adaptive to varying source statistics. The non-linear predictor adapts via an error feedback mechanism. In this adaptation process, CALIC only estimates the expectation of prediction errors conditioned on a large number of contexts rather than estimating a large number of conditional error probabilities. The former estimation technique can afford a large number of modeling contexts without suffering from the sparse context problem. The low time and space complexities of CALIC are attributed to efficient techniques for forming and quantizing modeling contexts.

1. INTRODUCTION

Recent years have seen an increased level of research in image compression. Most of the effort, however, has focused on the development of lossy compression techniques. Certain applications, such as medical imaging, image archiving, and remote sensing, require or desire lossless compression. Furthermore, lossless compression is often a necessary last step in many lossy image compression systems. Hence the development of efficient and effective lossless image compression is an important research topic that has many potential applications.

Statistical modeling of the source being compressed plays a central role in any data compression system. Fitting a given source well with statistical models is an ongoing and difficult academic endeavor pursued by researchers in a wide range of disciplines including source coding, machine learning, game theory, and estimation. A major difficulty with statistical modeling of continuous-tone images is the large alphabet size, typically 256 or larger. Context modeling of alphabet symbols (pixel values) leads to an intractably large number of possible model states (contexts). This is more than merely an implementation problem. If the number of model states is too large with respect to the size of the image, count statistics may not have enough samples to reach good estimations of conditional probabilities on the model states, leading to poor coding efficiency. This is known as "the sparse context" [9, 2] or "context dilution" problem [12]. The problem was theoretically formulated by Rissanen in the framework of stochastic complexity as the "model cost" [6, 7]. Rissanen's theoretical work proves that high complexity of a model can reduce coding efficiency, as observed by many data compression practitioners including the authors themselves [4, 13].

A central issue in the design of algorithmic techniques for lossless coding is the reduction of model cost leading to high coding and computational efficiency. In lossless image coding, the most common technique employed to reduce model cost has been the prediction/residual approach [3]. In such an approach prediction is used to capture the apriori smoothness of image data at low model cost. Prediction residuals are then modeled prior to encoding. CALIC employs this two step prediction/residual approach.

In the prediction step, CALIC employs a simple new gradient based non-linear prediction scheme called GAP (gradient-adjusted predictor) which adjusts prediction coefficients based on estimates of local gradients. Prediction is then made context-sensitive and adaptive by modeling of prediction errors and feedback of the expected error conditioned on properly chosen modeling contexts. The modeling context is a combination of quantized local gradient and texture pattern, two features that are indicative of the error behavior. The net effect is a non-linear, context-based, adaptive prediction scheme that can correct itself by learning from its own past mistakes under different contexts. The context-based error modeling is done at a low model cost. By estimating expected prediction errors rather than error probabilities in different modeling contexts, CALIC can afford a large number of modeling contexts without suffering from context dilution problem nor from excessive memory use. This is a key feature of CALIC that distinguishes it from existing methods.

Another innovation of CALIC is the way it distinguishes between binary and continuous-tone types of images on a local, rather than a global, basis. This distinction between the two modes is important because the compression methodologies are very different in the two modes. The former uses predictive coding, whereas the latter codes the pixel values directly. CALIC selects one of the two modes based on a local causal template without using any side information. The two-mode design contributes to the universality and robustness of CALIC over a wide range of images, including the so-called multimedia images that mix text, graphics, line art, and photograph.

The CALIC codec was proposed to ISO/JPEG as a candidate algorithm for the next international standard for lossless compression of continuous-tone images [14]. In the initial evaluation of all nine submitted proposals at the ISO/SC29/WG1 (JPEG/JBIG) meeting in Epernay, France, July 1995, CALIC had the lowest lossless bit rates in six of seven image classes: medical, aerial, pre-press, scanned, video, and compound document, and the third lowest bit rate in the class of computer generated images [11]. CALIC gives an average lossless bit rate of 2.99 bits/pixel on the 18 8-bit test images selected by ISO for proposal evaluation, versus an average bit rate of 3.98 bits/pixel for lossless JPEG (Huffman coding) on the same set of test images. Even more encouragingly, CALIC obtains a lower bit rate than the recent UCM (Universal Context Modeling) scheme proposed by Weinberger, Rissanen, and Arps [12]. The latter is a principled but highly complex context-based image coding technique, considered to be indicative of a lower bound on lossless bit rates achievable by other more practical methods.

Unlike UCM, CALIC is intended to be a practical lossless image codec. To attain this goal, new efficient algorithmic techniques have been developed for context formation, quantization, and modeling. Although conceptually more elaborate than many existing lossless image coders, CALIC is algorithmically quite simple, involving mostly integer arithmetic and simple logic. The prediction and modeling components of the encoder and decoder require only 1.52 CPU seconds on 512 × 512 continuous-tone images, and 2.28 CPU seconds on 720 × 576 continuous-tone images, when being executed on a SUN SPARC10 workstation. Both the encoding and decoding algorithms are suitable for parallel and pipeline hardware implementation while supporting sequential build-up. The CALIC codec is symmetric, meaning that the encoder and decoder have the same time and space complexities. The binary executables of CALIC are available for common hardware platforms [1].

In the rest of this paper we give a brief description of CALIC and its major components. For a detailed description the reader is referred to [13, 14]. We also present bit rates that CALIC obtained on all the ISO test images and make comparisons with some popular lossless image coders.

2. SYSTEM OVERVIEW

CALIC encodes and decodes images in raster scan order with a single pass through the image. The coding process uses prediction templates that involve only the previous two scan lines of coded pixels. Consequently, the encoding and decoding algorithms require a simple double buffer that holds two rows of pixels that immediately precede the current pixel, hence facilitating sequential build-up of the image.

CALIC operates in two modes: binary and continuous-tone modes. The binary mode is for the situation in which the current locality of the input image has no more than two distinct intensity values, hence it is designed for a more general class of images than the traditional class of black-and-white images. The system selects one of the two modes on the fly during the coding process, depending on the context of the current pixel. The mode selection is automatic, and completely transparent to the user. No side information about mode switching is required. In the binary mode, a context-based adaptive ternary arithmetic coder is used to code three symbols, including an escape symbol which triggers a return to continuous-tone mode. Values at the six nearest neighboring pixels are used as the context.

In the continuous-tone mode, the system has four major integrated components: prediction, context selection and quantization, context modeling of prediction errors, and entropy coding of prediction errors. We briefly describe each below.

2.1. GAP - Gradient Adjusted Predictor

GAP is a simple, adaptive, nonlinear predictor that can adapt itself to the intensity gradients near the predicted pixel. Hence it is more robust than the traditional DPCM-like linear predictors, particularly in areas of strong edges. GAP differs from the existing linear predictors in that it weights the neighboring pixels of $I[i,j]$ according to the estimated gradients of the image. In GAP the gradient of the intensity function at the current pixel I is estimated by computing the following quantities:

$$\begin{aligned} d_h &= |I[i-1,j]-I[i-2,j]| + |I[i,j-1]-I[i-1,j-1]| \\ &\quad + |I[i+1,j-1]-I[i,j-1]| \\ d_v &= |I[i-1,j]-I[i-1,j-1]| + |I[i,j-1]-I[i,j-2]| \\ &\quad + |I[i+1,j-1]-I[i+1,j-2]|. \end{aligned} \tag{1}$$

A prediction is then made by the following procedure:

IF $(d_v - d_h > 80)$ {sharp horizontal edge}
  $\dot{I}[i,j] = w$
ELSE IF $(d_v - d_h < -80)$ {sharp vertical edge}
  $\dot{I}[i,j] = n$
ELSE {
  $\dot{I}[i,j] = (w+n)/2 + (ne-nw)/4;$
  IF $(d_v - d_h > 32)$ {horizontal edge}
    $\dot{I}[i,j] = (\dot{I}[i,j] + w)/2$
  ELSE IF $(d_v - d_h > 8)$ {weak horizontal edge}
    $\dot{I}[i,j] = (3\dot{I}[i,j] + w)/4$
  ELSE IF $(d_v - d_h < -32)$ {vertical edge}
    $\dot{I}[i,j] = (\dot{I}[i,j] + n)/2$
  ELSE IF $(d_v - d_h < -8)$ {weak vertical edge}
    $\dot{I}[i,j] = (3\dot{I}[i,j] + n)/4$
}

where

$$\begin{aligned} n = I[i,j-1], w = I[i-1,j], ne = I[i+1,j-1], \\ nw = I[i-1,j-1], nn = I[i,j-2], ww = I[i-2,j]. \end{aligned} \tag{2}$$

The thresholds given in the above procedure are for eight bit data and are adapted on the fly for higher resolution images. They can also be specified by the user if off-line optimization is possible.

2.2. Coding Context Selection and Quantization

The predicted value $\dot{I}$ is further adjusted via an error feedback loop of one-step delay. This results in an adaptive,

[1] ftp://ftp.csd.uwo.edu/pub/from_wu

context-based, non-linear prediction $\ddot{I}$ as we will explain shortly. The residue obtained from the final prediction $\ddot{I}$ is entropy coded based on eight estimated conditional probabilities in eight different contexts. These eight contexts, called error energy contexts, are formed by quantizing an error energy estimator Δ, a random variable, into eight bins. The quantizer bins partition prediction error terms into eight classes by the expected error magnitude.

The error energy estimator is computed as follows:

$$\Delta = ad_h + bd_v + c \cdot |e_w|, \tag{3}$$

where d_h and d_v are defined in (1) and $e_w = I[i-1,j] - \dot{I}[i-1,j]$. The coefficients a, b, and c can be determined, in an off-line design process, by standard linear regression techniques. For algorithm efficiency, we recommend $a = b = 1$ and $c = 2$ in (3).

By conditioning the error distribution on Δ, we can separate the prediction errors into classes of different variances. Thus entropy coding of errors using estimated conditional probability $p(e|\Delta)$ improves coding efficiency over using $p(e)$. For time and space efficiency, we quantize Δ to L levels. In practice, $L = 8$ is found to be sufficient. Larger L will only improve coding efficiency marginally. Denote the Δ quantizer by Q, i.e., $Q(\Delta) \in \{0, 1, \cdots, 7\}$. In entropy coding of prediction errors, we estimate and use eight conditional probabilities $p(e|Q(\Delta))$. Since Δ is a random variable, it requires only scalar quantization. The quantization criterion is to minimize the total entropy of the errors. In an off-line design process, we get a training set of (e, Δ) pairs from test images, and use a standard dynamic programming technique to choose $0 = q_0 < q_1 < \cdots < q_{L-1} < q_L = \infty$ to partition Δ into L intervals such that

$$-\sum_{d=0}^{L-1} \sum_{q_d \leq \Delta < q_{d+1}} p(e) \log p(e) \tag{4}$$

is minimized. In practice, we found that an image-independent Δ quantizer whose bins are fixed as follows:

$$q_1 = 5, q_2 = 15, q_3 = 25, q_4 = 42, q_5 = 60, q_6 = 85, q_7 = 140$$

worked almost as well as the optimal image-dependent Δ quantizer.

2.3. Context Modeling of Prediction Errors and Error Feedback

Performance of the GAP predictor can be significantly improved via context modeling, because gradients alone cannot adequately characterize some of the more complex relationships between the predicted pixel $I[i,j]$ and its surrounding. Context modeling of the prediction error $e = I - \dot{I}$ can exploit higher-order structures such as texture patterns in the image for further compression gains. Hence, to further improve coding efficiency, we embed 144 texture contexts, into 4 error energy contexts ($\Delta/2$) to form a total of 576 so-called compound contexts. Texture contexts are formed by a quantization of a local neighborhood of pixel values to a binary vector. The vector of local values C is selected as follows:

$$\begin{aligned} C &= \{x_0, \cdots, x_6, x_7\} = \\ &\{n, w, nw, ne, nn, ww, 2n - nn, 2w - ww\}. \end{aligned} \tag{5}$$

C is then quantized to an 8-bit binary number $B = b_7 b_6 \cdots b_0$ using the prediction value $\dot{I}$ as the threshold, namely

$$b_k = \begin{cases} 0 & \text{if } x_k \geq \dot{I}[i,j] \\ 1 & \text{if } x_k < \dot{I}[i,j] \end{cases}, \quad 0 \leq k < K = 8. \tag{6}$$

Clearly, number B (a binary vector codeword) encodes the texture patterns in the modeling context which are indicative of the behavior of e.

Since the variability of neighboring pixels also influences the error distribution, we combine the quantized error energy $0 \leq \lfloor Q(\Delta)/2 \rfloor < L/2$ with the quantized texture pattern $0 \leq B < 2^K$ to form compound modeling contexts, denoted by $C(\lfloor Q(\Delta)/2 \rfloor, B)$. This scheme can be viewed as a product quantization of two independently treated image features: spatial texture patterns and the energy of prediction errors. At a glance, we would seemingly use $4 \cdot 2^8 = 1024$ different compound contexts, but careful analysis reveals that only 576 contexts are possible [13].

Since the conditional mean $\bar{e}(\delta, \beta)$ is the most likely prediction error in a given compound context $C(\delta, \beta)$, we can correct the bias in the prediction by feeding back $\bar{e}(\delta, \beta)$ and adjusting the prediction $\dot{I}$ to $\ddot{I} = \dot{I} + \bar{e}(\delta, \beta)$. In order not to over-adjust the GAP predictor, we actually consider the new prediction error $\epsilon = I - \ddot{I}$ rather than $e = I - \dot{I}$ in context-based error modeling. The context modeling of ϵ in turn leads to an improved predictor for I: $\ddot{I} = \dot{I} + \bar{\epsilon}(\delta, \beta)$, where $\bar{\epsilon}(\delta, \beta)$ is the sample mean of ϵ conditioned on compound context $C(\delta, \beta)$.

Besides aiding in obtaining an improved prediction, the estimated prediction error $\bar{\epsilon}(\delta, \beta)$ within each context is also utilized to sharpen the estimated conditional probabilities (hence reduce the underlying conditional entropies) that drive the entropy coder via a novel technique of sign flipping. This techniques works as follows: Suppose the current context is $C(\delta, \beta)$. Before encoding $\epsilon = I - \ddot{I}$, the encoder checks whether $\bar{\epsilon}(\delta, \beta) < 0$. If yes, $-\epsilon$, otherwise, ϵ is encoded. Since the decoder also knows $C(\delta, \beta)$ and $\bar{\epsilon}(\delta, \beta)$, it can reverse the sign, if necessary, to reconstruct ϵ.

2.4. Entropy Coding of Prediction Errors

The sign predicted prediction error is then encoded using the eight primary contexts Δ described above. An advantage of CALIC is the clean separation between the modeling and entropy coding stages. Any entropy coder, be it Huffman or arithmetic, static or adaptive, binary or m-ary, can be easily interfaced with the CALIC system.

The entropy coder in the binary mode has an alphabet size of three. Given this small alphabet size, we use a simple ternary adaptive arithmetic coder. Coding is done under 32 different contexts, for each of which we need to maintain a frequency table consisting of three symbols.

In the continuous-tone mode we used an adaptive m-ary arithmetic coder, CACM++ package that was developed and made publicly available by Carpinelli and Salamonsen. The software is based on the work by Moffat, Neal, Witten [5]. However, in order to improve coding and space efficiency some pre-processing of errors is done prior to entropy coding. Details are given in [14].

3. COMPRESSION PERFORMANCE

In order to demonstrate the compression performance of CALIC we compare it with some representative lossless image compression techniques. Table 1 lists compression results achieved with the set of ISO test images that were made available to the proposers. Column 2 lists the actual bit rates obtained by CALIC with the CACM++ implementation of arithmetic coding, Results with a static Huffman code are on an average 3.5% worse. Columns 3 and 4 list actual bit rates obtained by two publicly available lossless image compression codecs: the lossless JPEG implementation of Cornell University (LJPEG) and FELICS, a particularly simple lossless technique developed by Howard and Vitter [1]. The reported results of lossless JPEG were obtained by using the best of the eight JPEG predictors followed by Huffman coding of the prediction errors. FELICS uses context based prediction and Rice-Golomb coding of prediction errors. LJPEG and FELICS results on images with more than 8 bits per pixels are not given as the public domain implementations of these two only handle 8 bpp image data. Also, note that the images are mostly color images and have more than one color bands. Bit rates were obtained by appropriately averaging over the color bands weighted by size.

Both the current Lossless JPEG and FELICS are rather simple techniques that require minimal memory and computation resources. CALIC on the other hand is more complex and does require more resources, although the increase in memory and computation resources is modest. Hence we also compare CALIC with two other arithmetic coding based techniques, ALCM and CLARA that were proposed to ISO by UC Santa Cruz [8] and Mitsubishi [10] respectively. Both had memory requirements similar to CALIC and were of comparable complexity.

Image	calc	jpeg	felics	alcm	clara
air2	3.83	4.90	4.49	4.08	4.11
bike	3.50	4.33	4.06	3.69	3.63
cafe	4.69	5.63	5.31	4.99	4.86
tools	4.95	5.69	5.42	5.17	5.06
woman	4.05	4.84	4.58	4.30	4.15
cats	2.51	3.69	3.30	2.67	2.57
water	1.74	2.62	2.36	1.82	1.84
chart	1.28	2.23	2.14	1.27	1.36
graphic	2.26	2.81	2.85	2.41	2.24
faxballs	0.75	1.50	1.74	0.60	0.82
hotel	3.71	4.22	4.20	3.92	3.88
gold	3.83	4.22	4.21	4.02	3.88
finger	5.47	5.85	6.11	5.94	5.46
us	2.34	3.63	3.32	2.32	2.41
cmpnd2	1.24	2.50	2.40	1.34	1.47
cmpnd1	1.24	2.51	2.40	1.29	1.53
bike3	4.23	5.15	4.67	4.43	4.48
chart_s	2.66	3.86	3.44	2.77	2.88
mri	5.73	-	-	6.17	5.73
cr	5.17	-	-	5.43	5.22
xray	5.83	-	-	6.24	5.99
ct	3.63		-	4.09	4.08
aerial1	8.31	-	-	8.77	8.36

Table 1. Bit rates (bits per pixel) of CALC on ISO test set and comparison with a few selected schemes

REFERENCES

[1] P. G Howard and J. S. Vitter. Fast and efficient lossless image compression. In J. A. Storer and M. Cohn, editors, *Proceedings of the Data Compression Conference*, pages 351–360. IEEE Computer Society Pres, 1993.

[2] G. G. Langdon. Sunset: a hardware oriented algorithm for lossless compression of gray scale images. In *Medical Imaging V: Image Capture, Formatting and Display*, volume 1444, pages 272–282. SPIE, 1991.

[3] N. D. Memon and K. Sayood. Lossless image compression - a comparative study. In *Still Image Compression*, pages 8–20. SPIE Proceedings Volume 2418, 1995.

[4] N. D. Memon, K. Sayood, and S. S. Magliveras. Lossless image compression with a codebook of block scans. *IEEE Journal on Selected Areas of Communications*, 13(1):24–31, January 1995

[5] A. Moffat, R. Neal, and I. Witten. Arithmetic coding revisited. In *Proceedings of the Data Compression Conference*, pages 202–211, 1995.

[6] J. J. Rissanen. A universal data compression system. *IEEE Transactions on Information Theory*, 29(5):656–664, 1983.

[7] J. J. Rissanen. Universal coding, information, prediction and estimation. *IEEE Transactions on Information Theory*, 30:629–636, 1984.

[8] Don Speck. Proposal for next generation lossless compression of continuous-tone still pictures: Activity level classification model (ALCM). ISO Working Document ISO/IEC JTC1/SC29/WG1 N198, 1995.

[9] S. Todd, G. G. Langdon, and J. J. Rissanen. Parameter reduction and context selection for compression of gray scale images. *IBM J. Res. Develop.*, 29(2):188–193, 1985.

[10] I. Ueno and F. Ono. CLARA: Continuous-tone lossless coding with edge analysis and range amplitude detection. ISO Working Document ISO/IEC JTC1/SC29/WG1 N197, 1995.

[11] S. Urban. Compression results - lossless, lossy ±1, lossy ±3. ISO Working Document ISO/IEC JTC1/SC29/WG1 N281, 1995.

[12] M. J. Weinberger, J. Rissanen, and R. B. Arps. On universal context modeling for lossless compression of gray-scale images. Pre-print, 1994. To appear in the IEEE Transactions on Image Processing, 1996.

[13] X. Wu. Efficient and effective lossless compression of continuous-tone images via context selection and quantization. Pre-print, 1995. Submitted to the IEEE Transactions on Image Processing.

[14] X. Wu, N.D. Memon, and K. Sayood. A context-based, adaptive, lossless/nearly-lossless coding scheme for continuous-tone images. ISO working document ISO/IEC/SC29/WG1/N256, 1995.