A LOSSLESS IMAGE CODER WITH CONTEXT CLASSIFICATION, ADAPTIVE PREDICTION AND ADAPTIVE ENTROPY CODING

Farshid Golchin and Kuldip K. Paliwal*

School of Microelectronic Engineering
Griffith University
Brisbane, QLD 4111, Australia
F.Golchin, K.Paliwal@me.gu.edu.au

ABSTRACT

In this paper, we combine a context classification scheme with adaptive prediction and entropy coding to produce an adaptive lossless image coder. In this coder, we maximize the benefits of adaptivity using both adaptive prediction and entropy coding. The adaptive prediction is closely tied with the classification of contexts within the image. These contexts are defined with respect to the local edge, texture or gradient characteristics as well as local activity within small blocks of the image. For each context an optimal predictor is found which is used for the prediction of all pixels belonging to that particular context. Once the predicted values have been removed from the original image, a clustering algorithm is used to design a separate, optimal entropy coding scheme for encoding the prediction residual. Blocks of residual pixels are classified into a finite number of classes and members of each class are encoded using the entropy coder designed for that particular class. The combination of these two powerful techniques produces some of the best lossless coding results reported so far.

1. INTRODUCTION

Image coding is an essential component of many digital transmission and storage systems. Generally speaking, image coding algorithms can be divided into two categories: lossy and lossless. Lossy compression techniques are more popular, but they lead to coding distortions (or artifacts) which are not tolerable in certain applications. In areas such as medical image coding and some satellite imaging for instance, coding artifacts can have potentially adverse consequences or can corrupt data which has been obtained at a great cost.

It is in these areas that lossless image coding methods are utilized. Although lossless methods offer significantly lower compression ratios, the fact that they preserve the original data has made them indisposable. In recent years, a great deal of research in image coding has concentrated on lossy methods. The development of better transforms, quantizers and particularly adaptivity in coders has resulted in significant advances in this area. An adaptive coder is able to adapt its workings to the local characteristics within the different regions of an image.

Use of "context" [1] information to provide adaptivity has demonstrated significant improvements in lossless compression results. However, many of these techniques are based on heuristics and often do not fully exploit the gains offered by adaptive coding.

*The first author is supported in part by a CSIRO Division of Telecomm. and Industrial Physics postgraduate scholarship.

A lossless image coder in its simplest form is implemented using a fixed predictor and a fixed entropy coder. The predictor makes a prediction for each pixel based on previously transmitted values. The predicted values are then subtracted from the original values, thus leaving the prediction error (or residual). An entropy coder is then used to encode the residual signal.

Adaptivity can be provided by making the predictor and/or the entropy coder adaptive. An adaptive predictor is able to adapt to the characteristics of various textures or the direction of edges and hence produce more accurate predictions. An adaptive entropy coder, on the other hand, can adjust its workings to better match the local statistics of an image and produce shorter-length codes.

2. CONTEXT CLASSIFICATION AND ADAPTIVE PREDICTION

In this paper, we concentrate on one particular type of adaptive prediction which is performed by switching among a finite number of predictors. The decision of which predictor to use is made for each pixel being encoded and is solely based on previously transmitted pixel values.

The difficulty with this type of adaptive prediction schemes is in the trade-off that exists between the number of predictors and computational complexity. Since the appropriate predictor is often selected using an exhaustive search [2] the computational complexity is high and can only be reduced by limiting the number of predictors. We have previously presented an optimal scheme for the design of a small number of predictors [2]; however, this scheme requires a computationally intensive design process.

The main performance obstacle in the path of the adaptive prediction schemes mentioned above is the exhaustive search required to map the local context from pixel domain to an appropriate predictor. The performance can be significantly improved if a faster mapping is used. Such a mapping has been recently proposed by Wu and Memmon [6] [5]. The coder named CALIC [5], has many distinct features which contribute to its state-of-the-art performance; however its most significant (and possibly understated) feature is its Context Classification (or Quantization).

In CALIC, a fast method is used to classify the causal neighborhood (context) of the pixel to be encoded into a finite number of contexts. For each of the possible contexts, a bias value (a scalar) is maintained which is added to the prediction made by a less sophisticated linear predictor in order to improve its performance. CALIC assumes that the predictor is consistently repeating a similar prediction error over the same context which can be compen-

sated for by adding the bias value.

In this paper, we take a slightly different approach and set out to exploit this fast mapping to the fullest. We do so by using context classification to select a linear predictor from a set of 1024 linear predictors. In this fashion, a pixel-by-pixel decision is made as to which linear predictor should be used for predicting that particular pixel.

As done in [6], we define the prediction context of a pixel X (see Fig. 1) as:

$$C(X) = (w, ww, nw, n, nn, ne, 2n - nn, 2w - ww). \quad (1)$$

where each of the elements is the pixel in that direction (north, south, ...) with respect to the pixel being encoded. The two values $2n - nn$ and $2w - ww$ do not correspond to actual pixel values. These values are calculated from the values of pixels n,nn,w and ww and are used to provide some measure of the gradient of intensity within the context. For the purpose of notation, we also index the values in $C(X)$ such that $C_1(X) = w$, $C_2(X) = ww$, ... , $C_8(X) = 2w - ww$.

Next, we calculate the value μ which is the mean of all pixels in the context of X:

$$\mu = \frac{1}{8} \sum_{i=1}^{8} C_i(X). \quad (2)$$

Using the value μ as a reference point, we are now in a position to generate the 8-bit, binary string $S(X)$ which defines the shape or the texture of the context:

$$S(X) = (S_1(X), S_2(X), ..., S_8(X)). \quad (3)$$

where,

$$S_i(X) = \begin{cases} 0, & \text{if } C_i(X) \geq \mu, \\ 1, & \text{otherwise.} \end{cases} , for\ i = 1, 2, ..., 8. \quad (4)$$

The 8-bit binary string $S(X)$, can also be interpreted as a binary number ranging between 0 and 255. We use this number as an index for the classification of the context $C(X)$. This index classifies each context according to its texture, gradient, edges and so on. This type of classification is arguably the key factor in the success of CALIC. However, rather than using the Gradient Adjusted Predictor (GAP) used in CALIC to produce a reference value, we have used the mean value μ which is easier to calculate.

So far, the shape of the context has been classified into one of 256 different shapes. However, since in calculating $S(X)$, all magnitude information has been discarded and only sign information has been retained, $S(X)$ does not contain any information about the strength of textures, steepness of gradients or the sharpness of edges. In order to incorporate some of this information into the context classification, we also calculate a separate index corresponding to the level of activity (or energy) within the context.

To calculate the energy index, we first need to define some measure of the energy within the context. We do so by calculating the standard deviation of the values in each context:

$$\sigma = \sqrt{\frac{1}{8} \sum_{i=1}^{8} [C_i(X) - \mu]^2}; \quad (5)$$

The standard deviation value σ is then quantized into one of 4 levels using Lloyd-Max [8] scalar quantizer. This quantizer is designed once using data from a set of training images and is kept for all future usage. In this fashion, the quantization index $q(X) \in \{1, 2, 3, 4\}$ is found for the context of each pixel to be predicted. This index provides the activity information which is not contained in the index $S(X)$ calculated previously. The two indices complement each other in terms of information and their combination is used to produce an effective classification for each context:

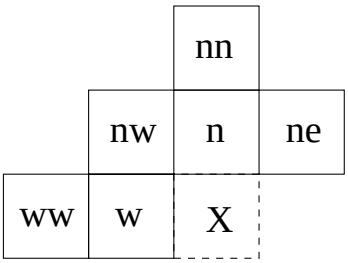

Figure 1: The Prediction context of pixel X

$$I(X) = q(X)\ S(X). \quad (6)$$

In this way, $I(X)$ becomes an integer between 0 and 1023 which is used as the final classification index for each context. We design an optimum 4-th oder linear predictor for each of the 1024 contexts using either training data or the image to be encoded. In coding, the context of each pixel is classified and then the appropriate predictor is used for the prediction of that particular pixel. It should be noted that the predictors for each context are designed using a Mean-Squared-Error (MSE) criterion.

3. THE MINIMUM-ENTROPY CLUSTERING ALGORITHM

The adaptive prediction scheme described in the previous section, decorrelates neighboring pixels in the image. However, upon inspecting the residual image, it becomes clear that large values of residuals (prediction error) tend to be confined to certain areas while small residual values are also grouped together.

This non-uniformity in the local statistics of the areas within the residual image can be exploited in entropy coding. This is done through using multiple entropy coders, each of which is matched to a particular type of region in the residual image. The question remains as to how the different areas within the image can be classified into different types and how the entropy coders can be designed. To this end, we utilize the Minimum-Entropy Clustering (MEC) [9] algorithm.

Our aim in the use of the Minimum-Entropy Clustering algorithm is to classify blocks of samples and then encode the samples in each block using a suitable entropy coder. The classification and design of the entropy coders should be performed in a way such that the overall entropy is minimized.

In order to design a coding system with N entropy coders, the MEC algorithm operates as follows:

1. *Initialization*: N probability distribution functions (PDF's) are defined. These PDF's will define the initial classes.
2. *Minimum-Code-Length classification*: Each block of samples $B = (b_0, b_1, ..., b_{m-1})$ is classified as belonging to

class C such that the code length (after entropy coding) $L = -\sum_{i=0}^{m-1} log_2\ p(b_i|C)$ is minimized. This is similar to a nearest neighbor selection in VQ design.

3. *Re-Estimate class statistics*: Estimate the new class PDF's. This will ensure that the class statistics are matched to those of the samples in that class and hence the entropy is further reduced. This step is similar to a centroid calculation in VQ design.

4. *Iteration*: Stop if a maximum number of iterations is reached or the classes have converged. Otherwise, go to step 2.

Steps 2 and 3 form the core of the MEC algorithm. In each of these two steps, the overall entropy is reduced. There are a number of choices available for the initial classification. However, a better definition of the initial classes can reduce the number of iterations required. Since we wish to group together blocks with similar statistics, a reasonable choice for initial classification would be classification based on the variance of the blocks. To do so, we choose a classification similar to that used by Chen and Smith [3]. The variance of each block is estimated and the blocks are sorted in the order of increasing (or decreasing) variance.

Using the sorted list, $m-1$ threshold values (of variance) are selected and used to classify the blocks into m classes. After this classification, the class PDF's are estimated and used to define the initial classes.

This algorithm may be used in either a parametric or a non-parametric form. In its parametric form, all distributions are modeled as Generalized Gaussian (GG) distributions [4] whose shape parameter and variance are estimated. In this case, step 3 of the algorithm becomes a maximum-likelihood estimation of the shape parameter and variance. The main disadvantage of using the parametric form of the MEC algorithm is the additional processing required in the calculation of probabilities (Step 2) and the parameter estimation (Step 3).

In the non-parametric version of the MEC algorithm, frequency tables are maintained for each of the classes. These frequency tables are updated in step 3 of the algorithm. After the MEC algorithm has converged, these frequency tables are used to design the entropy coders. It is for this reason that the frequency tables must also be known at the decoder. In the parametric case, the probabilities can be efficiently described to the decoder by transmitting two parameters (shape and variance) per class.

In the non-parametric version, the frequency tables must be explicitly transmitted. To reduce the amount of information, we may take advantage of the symmetry in the frequency tables. One half of each frequency table is quantized using 6-8 bits and then transmitted to the decoder. If adaptive entropy coders are used, the probability tables are quickly adapted to match the source statistics and any mismatches caused by quantization rapidly disappear.

There is a trade-off between two versions of the MEC algorithm mentioned above. The non-parametric version offers faster execution times at the expense of added side-information; although the parametric version is slower to run it is much more concise to transmit its parameters. However, in terms of compression the two versions of the MEC algorithm produce quite similar results. The results in this paper have been obtained using the non-parametric version of the MEC algorithm.

4. A CONTEXT CLASSIFICATION BASED IMAGE CODER

In this section, we examine the design of the adaptive predictor and the entropy coders used to encode the prediction residuals. As mentioned in the previous section, the predictors can be designed either for a training set of images or specifically for the image to be encoded. There is a trade-off between the the two methods which parallels the trade-offs experienced in quantizer design.

If a training set of images are used, the training is performed off-line, however the predictors are not designed for the particular image and the predictors are slightly inferior in performance. Alternatively, if the predictors are designed for the image to be encoded, then some online training is required and the predictor coefficients must be transmitted to the receiver.

In this paper, we examine both of the above-mentioned scenarios and compare their performance. The training algorithm for the predictors is as follows:

1 For each pixel in the image (Training image, or image to be encoded) the context $C(X)$ is found.

2 We calculate and remove average value μ from the pixels in $C(X)$.

3 Take the sign of each of the mean removed values as +=0 and -=1 to make an 8-bit binary string $S(X)$.

4 Calculate the standard deviation of the values in $C(X)$.

5 Quantize the Standard deviation into one of 4 levels to find energy index $q(X)$.

6 The pixel is classified as belonging to 1024 contexts by combining $S(X)$ and $q(X)$.

7 Update autocorrelation values for that context.

8 Return to Step 1 (until the entire image has been processed)

The standard deviation values are quantized using a Lloyd-Max quantizer [8]. Once the classification has been completed and the autocorrelation values are finalized, the autocorrelation values are used to design a 4th-order optimum linear predictor for each context. If a particular context appears rarely or not at all (in the training set), then it is allocated a trivial second order predictor with two coefficients of 0.5.

It should also be noted that if the predictors are specifically designed for the image to be encoded, their coefficients must be transmitted as side information and contribute around 0.01 to 0.02 bpp to the overall bit-rate. All predictor coefficients are quantized using 10 bits per predictor coefficient.

It is interesting to note that within a typical image, only 100-300 of the possible 1024 contexts appear. This is a significant advantage with respect to the transmission of the predictor coefficients, since only a small portion of the predictor coefficients need to be transmitted.

The MEC algorithm was used to define a classification scheme and design the entropy coders. Classification was made on blocks of 8x8 pixels since it was experimentally found to produce the best results. The 8x8 blocks were classified into 16 classes and hence encoded using 16 different entropy coders. As mentioned previously, along with the classification scheme, the frequency tables for the entropy coders must also be transmitted to the decoder. For a 512x512 pixel image this results in an added bit-rate of 0.01 bpp. For a 256x256 pixel image this overhead increases to approximately 0.04 bpp.

Coded Image	Coding Method		
	CCBAC-TS	CCBAC-SI	CALIC [6]
barb	4.34	4.29	4.43
fruit	4.42	4.36	4.55
lena	3.94	3.90	4.05
man	4.38	4.35	4.43
goldhill	4.70	4.64	4.72

Table 1: A comparison of lossless compression results (quoted in bpp)

5. RESULTS AND CONCLUSIONS

The coding results are listed in Table 1. The Context Classification Based Adaptive Coder trained on a Training Set is referred to as CCBAC-TS and the coder trained on the Specific Image is referred to as CCBAC-SI. A set of ten images, excluding the training set was used for training the CCBAC-TS. The values quoted in the table are based on actual file sizes and include all overheads.

The results obtained from CALIC [6], were also provided for comparison. From Table 1, it is clear that both proposed methods significantly outperform CALIC which is considered to be the state-of-the-art in lossless image coding.

As expected, the image specific training algorithm CCBAC-SI outperforms the pre-trained predictors used in CCBAC-TS. What is surprising however, is the small difference between the two sets of results. This suggests that in many cases, the pre-trained predictors may be advantageous, since they require no "online" training.

The use of "off-line" training (using a training set of images), is even more appropriate when the coder is being used on satellite or medical images. For instance, if the coder is going to be used in encoding X-ray images, then the predictors can be trained using a suite of X-ray images.

Both of the tested coders outperform CALIC. We must, however, acknowledge the increased computational complexity of both coders compared to CALIC. Work is currently in progress toward further optimizing the speed of the presented algorithms.

The prediction scheme presented in this paper, demonstrates the advantages of using multiple predictors and the fast context classification scheme presented. The fact that the context classification scheme does not use an exhaustive search allows a large number of contexts to be utilized. The combination of this type of prediction with an adaptive entropy coding scheme such as the MEC algorithm was shown to produce some of the best results obtained in image coding to date.

6. REFERENCES

[1] T.V. Ramabadran and K. Chen, "The use of contextual information in the reversible compression of medical images," *IEEE Trans. Medical Imaging*, vol. 11, pp. 185-195, June 1992.

[2] F. Golchin and K.K. Paliwal, "Clustering-based adaptive prediction and entropy coding for use in lossless image coding," *Accepted for ICIP-97*, October 1997.

[3] W.H. Chen and C.H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. on Commun.*, vol. COM-25, pp. 1285-1292, November 1977.

[4] N. Farvardin and J.W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485-497, May 1984.

[5] X. Wu, "Context selection and quantization for lossless image coding," *Proc. Data Compression Conf. '95*, March 1995.

[6] X. Wu and N. Memmon, "CALIC - A context based lossless image codec," *Proc. Int. Conf. on Acoust., Speech, Signal Processing*, Atlanta, GA, pp. 1891-1894, May 1996.

[7] Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, COM-28, pp. 84-95, January 1980.

[8] S.P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, IT-28, pp. 127-135, March 1982.

[9] F. Golchin and K.K. Paliwal, "Minimum-entropy clustering and its application to lossless image coding," *Accepted for ICIP-97*, October 1997.