

A Hybrid Lossless Compression of Still Images Using Markov Model and Linear Prediction

Seishi TAKAMURA and Mikio TAKAGI

Institute of Industrial Science, The University of Tokyo

Abstract. Markov model approximation of digital images gives very low entropy, but requires a huge amount of storage to maintain the frequency table for encoding and decoding. Therefore, it has only been used for binary picture compression so far.

We have developed a high performance DPCM-based predictive lossless image coder, and have combined this technique with a Markov model, which provides a better compression ratio for gray scale images.

The basic idea is, encode the first part (this size is variable) of an image using the normal DPCM-based encoder and concurrently use these data as a Markov model training set. Then the latter half will be encoded using the Markov model with the assistance of the DPCM coder.

1 Introduction

Lossless still image compression techniques have recently been highlighted. The most popular method for lossless compression of still images with more than 8 bits/pixel is DPCM-based predictive coding like JPEG lossless mode, but performance limitations are beginning to appear.

On the other hand, the Markov model entropies of the images are attractively lower than the bit rate of predictive coders (e.g. table 1 in section 4), but a frequency table with a huge amount of data is necessary for encoding and decoding, therefore it has been unrealistic to apply the Markov model for image compression up to the present.

In this paper, we propose using a Markov model for image compression and show that the performance of current lossless image compression techniques can be improved.

Our method consists of Markov model encoding and classical DPCM-based encoding. It does not require the transmission of a huge frequency table. In addition, our DPCM-based coder itself outperforms the JPEG lossless coder.

2 Markov Model Image Representation for Image Coding

If the symbol occurrence probability depends only on the m preceding symbols, the source is called an “ m -th order Markov source”. The series of m preceding symbols is called the “state”. If the source has N different kinds of symbols, the m -th order Markov model can be divided into N^m memoryless sources, the same as the number of possible states.

To encode such a Markov source, N^m different occurrence frequency tables are necessary. As each table has N words, total table size amounts to N^{m+1} words. At the moment, current techniques using the Markov model only treat binary ($N = 2$) images

for $m \simeq 10$. Needless to say, it is impractical to transmit the whole table for gray images ($N \simeq 2^8 = 256$) because of its astronomical size.

We use a 2-dimensional second order Markov model whose state S is represented as $S(X_1, X_2)$, where X_1 and X_2 are as shown in figure 1. Although this model is very simple, the entropy of this model is much lower than the bit rate of former lossless compression scheme (see table 1), which implies further compression possibilities.

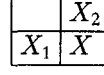


Fig. 1. Support of a second order Markov model

2.1 Learning Markov model

First of all, we encode the first image part (it is variable in size) of an image with a certain lossless coder, and use these image data for the training of the Markov model. After the training we try to encode the latter part with the Markov encoder (see the following section for details). The coder is also trained while it encodes. Therefore there is no need to transmit the frequency table of each Markov state.

Appearance of an unknown symbol In compensation of no table transmission, a key question is how to deal with in the coding process of the latter image part. It is “the appearance of an unknown symbol s ”, because the decoder has no means of detecting it. Therefore some means of informing the decoder must be provided.

To solve this problem, at the moment we prepare the special signal “Unknown”. Each frequency table prepares room for this signal. In encoding, when the coder detect the new symbol, “Unknown” is transmitted. Suppose A is the fixed frequency for this symbol, “theoretical code size for the model” H_A is

$$H_A = \sum_S \sum_{s \in S} -\#(X) \log_2 \frac{\#(s)}{\#(S) + A} \quad (\text{bit}), \quad (1)$$

where S is a state, s is a symbol and $\#(\cdot)$ means the number of the set. Note that $H_A|_{A=0}$ is equal to the Markov model entropy. If the state S is unknown or the symbol is unknown, we use the normal prediction coding methods using the neighbor pixels (see section 3) to recover the value of current pixel.

We use arithmetic coder which is suited to encode dynamically-varying probability fractions with high coding efficiency. The evaluation of the other trial of letting the decoder know when a new symbol appears and the comparison with JBIG will also be shown in subsection 4.1.

3 DPCM-based Encoding Algorithm

In this section we explain our DPCM-based predictive coder which is used together with the Markov model image coder. The scheme consists of two main parts: pixel classification and source distribution fitting.

3.1 Pixel classification

We use two methods to classify each pixel. The first method is used to decide the *group* of each pixel and calculate the prediction coefficients respectively, and the second method

calculates the *class* for each pixel, which is used to split the data stream of prediction error and to reduce entropy.

An important property of these methods is they can be done using only past data. Therefore the encoder have to transmit no information about *group* and *class* since the decoder can generate the same information independently.

Grouping according to local edge Figure 2 shows X's neighbor pixels $a \sim f$ used to determine the group it belongs to.

The group number of X is decided according to which number among $(|a - f|, |b - f|, |c - f|, |e - f|, |f + c - 2b|, |b - c|, |d - c|)$ is the biggest. This method enables X to be distinguished according to its local edge direction. We calculate the prediction coefficients using the least square error method for each group respectively.

a	b	c	d
e	f	X	

Fig. 2. Neighbor pixels $a \sim f$ to decide the group of pixel X

Classification according to the statistical property[4] It is also possible to classify a pixel according to the value $Q = |e - f| + |b - f| + |c - f| + |d - f|$, which physically means the "edge intensity around X". After obtaining this Q value, it is quantized properly and used as *class* number. Since this procedure is independent from the former classification method, they can be used together.

Effect of pixel grouping For example, Gaussian sources with different variances should not be entropy coded together, but separately, because the latter always gives a shorter code. Therefore the source splitting improves the performance of compression if it is successful.

Our classification method is not directly related to the source property itself. But from the experimental results, the source is split into different statistical properties, which means an improvement in compression.

3.2 Improvement of coding efficiency

To encode signals effectively, a knowledge about its distribution is necessary. However sometimes the size of the table becomes significant compared to the code size. In our method, we find the best fit function of the error distribution and then transmit the parameters, which has negligible overhead so that the decoder can re-generate an approximation of the frequency table. Since the fitting curve is not perfect, the coder updates the internal table after encoding/ decoding each symbol, which is described next section.

Error conversion Normally the prediction error is rounded off to an integer and encoded. In addition to this simple rounding, we also try to convert the prediction error in the following way and examine its entropy to obtain a smaller code.

First we obtain the upper and lower bound of a group (max, min). Then convert the actual pixel value into a non negative integer (figure 3). (In this figure, if the actual pixel is equal to 'max', $E = 9$.) This conversion is obviously reversible.

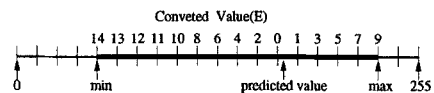


Fig. 3. Concept of error conversion

Criterion of fitting Giving consideration of subsequent entropy coding, the criterion of “better fitting” should be considered. We use “the theoretical information amount I ” according to the probability function, i.e.:

$$I = \sum_s -\#(s) \log_2 f(s) \quad (\text{bits}), \quad (2)$$

where $\#(s)$ means the number of occurrences of symbol s and $f(s)$ is the fitting function which satisfies $\sum_s f(s) = 1$.

Error distribution fitting To find well approximated fitting, some different fitting curves, such as (Generalized) Gaussian function and Spline function (cubic natural spline) are attempted. Generalized Gaussian function with zero mean $f_{p,\sigma}(s)$ is represented as $p \exp(-|s/(\gamma\sigma)|^p) / (2\Gamma(1/p)\gamma\sigma)$, where $\gamma = \sqrt{\Gamma(1/p)/\Gamma(3/p)}$ and σ^2 is the variance of s . Sometimes this distribution do not fit the actual distribution well because of its asymmetry. This is the reason why spline function is also tried.

Both simple prediction error distribution and E (converted value) distribution are tried to fit by these functions and the best one according to the criterion equation (2), which also considers the *overhead*, is chosen. This *overhead* is the number of the bits for each fitting function to represent its parameters such as node coordinates of spline, σ and p for $f_{p,\sigma}$. An example of fitting of a certain distribution is shown in figure 4. In this case, the generalized Gaussian distribution for E is chosen.

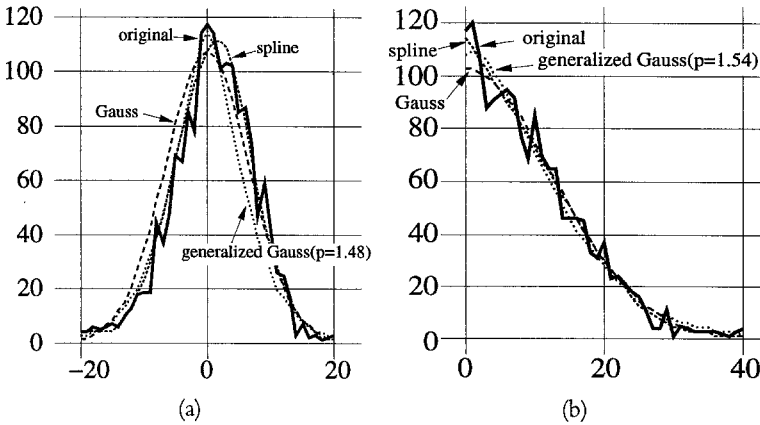


Fig. 4. The fitting for the error distribution. (a): rounded prediction error (b): converted error E .

4 Experimental Results

The compression performance of our coder with varying training sizes is tested on several gray-scale images and compared with other lossless compression schemes. We used 24 neighbor pixels for DPCM-based linear prediction, the results are shown in table 1. The percentage “0%” means trying to compress all the image with Markov encoder except

the points where the new states or new symbols appear. “100%” means whole-DPCM encoding.

The smallest bit-rate for each image is underlined. The first 5 images are 720×576 pixels, “Washdc1..7” are multi-band LANDSAT images with 512×512 pixels.

Table 1. Entropy, bit rate of JPEG lossless coder and our method for each image(unit = bit). **0th E:** 0th order entropy, **M.M.E:** Second order Markov model entropy, **JPEG:** Bit rate of JPEG Lossless coder (by ‘crush’ [6]). The numbers inside the parentheses is the predictor number, **percentage:** Ratio of the first DPCM-encoded trainig part. **overhd:** Overhead size (bytes) for noticing the occurrence of new symbols, **JBIG:** Code size (bytes) for informing the same information by the JBIG coder.

Image	0th E.	M.M.E.	JPEG	0% (overhd:JBIG)	50% (overhd:JBIG)	100%
girl	6.608	3.079	4.981(8)	<u>3.653</u> (23396:25571)	4.136 (10589:11771)	4.861
boats	7.088	3.300	4.185(8)	4.465 (45978:35521)	4.371 (34060:22798)	<u>4.107</u>
goldhill	7.530	3.675	4.653(8)	4.943 (62949:46104)	4.720 (27073:20430)	<u>4.450</u>
zelda	6.741	2.990	4.768(8)	<u>3.501</u> (20151:22735)	4.091 (7518:8551)	4.770
hotel	7.546	3.320	4.667(8)	<u>4.719</u> (29953:41969)	4.582 (14960:18471)	<u>4.352</u>
washdc1	4.685	3.442	3.849(7)	3.748 (9074:10309)	3.652 (2989:3588)	<u>3.543</u>
washdc2	3.846	2.726	2.981(7)	2.881 (4107:5402)	2.817 (1249:1802)	<u>2.738</u>
washdc3	4.307	3.153	3.484(7)	3.390 (7105:8357)	3.286 (2330:2842)	<u>3.183</u>
washdc4	5.432	3.909	4.324(7)	4.434 (20084:18670)	4.206 (7315:7052)	<u>4.007</u>
washdc5	5.681	4.056	4.649(7)	4.816 (27274:22968)	4.567 (10226:9097)	<u>4.351</u>
washdc6	3.441	0.903	0.920(4)	0.948 (413:712)	<u>0.894</u> (171:334)	0.938
washdc7	4.757	3.454	3.792(7)	3.747 (9652:10825)	3.617 (3141:3721)	<u>3.511</u>

4.1 Representation of new symbol appearance

To substantiate our method to deal with the occurrence of new symbols, the theoretical overhead of the extra information necessary for representing the occurrence of new symbol according to equation (1) is calculated. It is also possible to inform the decoder about the new symbol by encoding the binary figure like figure 5(b), since with such binary figures the decoder can know where the new symbol occurs. We adopt the JBIG coder to compress such data.

Figure 5(a) shows the original image “zelda” and the black pixels in figure 5(b) shows the occurrence points of the unknown symbols. The first half (50%) of the image is used for the initial training. There are 12551 occurrences within the latter half (207360 pixels). This image is compressed to 8551 bytes by JBIG coder while the calculated overhead of our method is 7518 bytes. Other results are shown within the parentheses in table 1. Sometimes the JBIG coder outperforms our method in regard to new symbol treatment.

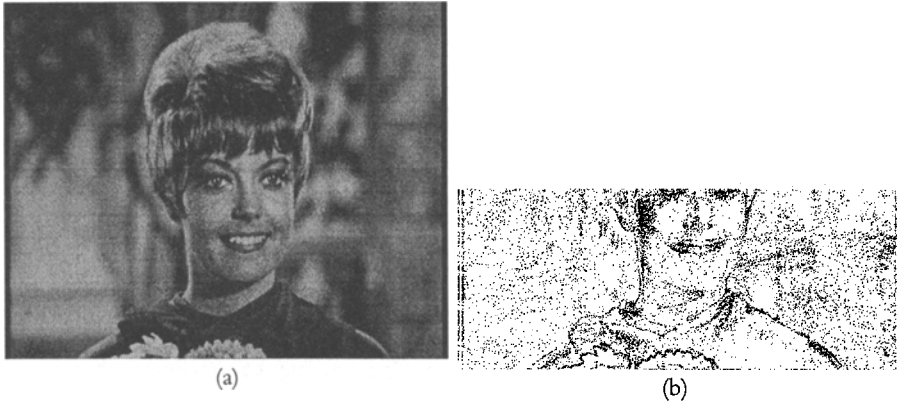


Fig. 5. (a) Sample image “zelda”. (b):The positions of the new occurrence of the symbols in image (black dots).

5 Conclusion and Future Work

In this paper, the algorithm for continuous tone still image compression using a Markov model is proposed. The coder always outperforms JPEG lossless coding schemes by 0.2~1.3 bits/pixel.

In our method, the predictive coding part and the Markov model coding part are independent, therefore it is possible to combine more powerful predictive coders.

Although it is clear for some images (girl, zelda) that the Markov model coding improves the performance, there are still some images for which it does not work well. We will further investigate the following points:

- efficient treatment of the appearance of the unknown symbol
- investigate the Markov state reduction
- application for the color image coding and near-lossless coding

References

1. S. Takamura et al.: “Lossless Image Compression with Lossy Image using Adaptive Prediction and Arithmetic Coding”, Proc. Data Compression Conference '94, pp. 166–174, 1994
2. D. M. Abrahamson: “An Adaptive Dependency Source Model For Data Compression”, Comm. ACM, Vol. 32, No. 1, pp. 77–83, Jan. 1989
3. P. G. Howard, J. Vitter, “New Methods for Lossless Image Compression Using Arithmetic Coding”, Proc. Data Compression Conference, Apr. 1991, pp.257–266
4. T. Taniguchi et al.: “Variable-Length-Code-Selective Reversible Predictive Coding for Multi-Level Images”, Trans. Institute of Electronics, Information and Communication Engineers (in Japanese), Vol.J70-B, pp.654–663, Jun. 1987
5. S. Utsui et al.: “Study on a Lossless Coding of Natural Images”, Proc. 9th Picture Coding Symposium of Japan (in Japanese), pp.103–104, Oct. 1994
6. ‘crush’ package is available via anonymous ftp at [dftnic.gsfc.nasa.gov](ftp://dftnic.gsfc.nasa.gov/disk$moie:[anonymous.files.software.unix.crushv3]:crush.v3_tar.Z):
disk\$moie:[anonymous.files.software.unix.crushv3]:crush.v3_tar.Z