Predictor Order and Error Distribution of MMAE Predictors for Lossless Image Coding

Hirokazu UENO[1] and Yoshitaka MORIKAWA[2]
[1]Graduate School of Natural Science and Technology, Okayama University
3-1-1, Tsushima-naka, Okayama, Okayama 700-8530, Japan
[2]Graduate School of Natural Science and Technology, Okayama University
3-1-1, Tsushima-naka, Okayama, Okayama 700-8530, Japan
E-mail : [1]ueno@trans.cne.okayama-u.ac.jp, [2]morikawa@cne.okayama-u.ac.jp

Abstract: This paper investigates the relation between an error distribution and a predictive order of minimum mean absolute error predictors(MMAE predictors) designed for lossless coding of greyscale images. Design of MMAE predictors reduces to the linear programming problem. Let k is the number of coefficients in a predictor(predictor order), we imagine that predictor order k have a distribution shaping effect. Main purpose of this paper is to ensure that k have such an effect.

1. Introduction

Recent years have seen an increased level of research in lossless image compression, in addition to lossy compression. Lossless image codings are required and desired in certain applications such as medical and satellite imagings and digital archiving of cultural heritages. Since the predictive coding scheme enables us to predict each pixel one by one and rather precisely in aid of adaptation, many lossless coding schemes employ prediction.

For lossless image coding based on prediction, the coding performance depends largely on the efficiency of predictors. Many lossless image coding formats use minimum mean square error predictors(MMSE predictors)[1][2][3], but MMSE predictors are susceptible to edges(the big transition part of the brightness value) in images. So Hashidume et al.[4] have proposed minimum mean absolute error predictors(MMAE predictors)[4] which are robust to edges. [4] says that using MMAE predictors the accuracy of prediction is enhanced and entropy of prediction is reduced.

This paper investigates the relation between an error distribution and a predictor order of MMAE predictors designed for lossless image coding. Design of a MMAE predictor reduces to the linear programming problem. Let k is the number of coefficients in a predictor(predictor order), at least k prediction errors become 0. Thus, we imagine that predictor order k have a distribution shaping effect. Main purpose of this paper is to ensure that k have such an effect.

In this paper, we define the fitting degree of actual prediction error distribution and modeling distribution(laplace distribution) in terms of the redundancy(the difference of entropy and model entropy), and measure the redundancy changing the value of k. Then we found that the optimum k appears to minimize the redundancy.

2. Minimum Mean Absolute Error Predictor

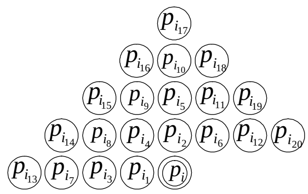


Figure 1. Pixels of support region ($k = 20$).

When we denote the current pixel p_i's value $B(p_i)$, the predicted value $\hat{B}(p_i)$ is calculated by

$$\hat{B}(b_i) = \boldsymbol{\theta}_i^T \cdot \boldsymbol{a}, \tag{1}$$

where $\boldsymbol{\theta}_i = [B(p_{i_1}), B(p_{i_2}), \cdots, B(p_{i_{20}})]^T$ is the local causal area(support region) vector of p_i (see Figure 1) and $\boldsymbol{a} = [a_1, a_2, \cdots, a_{20}]^T$ is the vector of prediction coefficients for p_i. When we denote the set of pixels in coding area $\boldsymbol{R} = \{p_i | i = 1, 2, \cdots, S\}$, a problem to design a MMAE predictor for $\boldsymbol{R}$ can be written as a mathematical programming problem as follows:

$$\begin{aligned}
&\underset{\boldsymbol{a}}{\text{Minimize}} \quad \| \boldsymbol{e} \|_1 = \sum_{p_i \in \boldsymbol{R}} |e_i| \\
&\text{subject to} \quad \boldsymbol{e} = \boldsymbol{B} - \hat{\boldsymbol{B}}, \\
&\qquad \boldsymbol{e} = [e_1, e_2, \cdots, e_S]^T, \\
&\qquad \boldsymbol{B} = [B(p_1), B(p_2), \cdots, B(p_S)]^T, \\
&\qquad \hat{\boldsymbol{B}} = [\hat{B}(p_1), \hat{B}(p_2), \cdots, \hat{B}(p_S)]^T, \\
&\qquad \hat{B}(p_i) = \boldsymbol{\theta}_i^T \cdot \boldsymbol{a} \ \ (for\ i = 1, 2, \cdots, S), \\
&\qquad \boldsymbol{a} = [a_1, a_2, \cdots, a_{20}]^T.
\end{aligned} \tag{2}$$

The absolute part of the objective function makes this problem difficult to solve. So, we transcribe the ith element of the vector $\boldsymbol{e}$ as

$$e_i = e_i^+ - e_i^-, e_i^+ \geqq 0, e_i^- \geqq 0, \tag{3}$$

and boil down this problem to a linear programming problem. In the same way, transcribing a_j which is the jth element of the prediction coefficients $\boldsymbol{a}$, the problem (2) can be rewritten as a linear programming problem as follows:

$$
\begin{aligned}
\underset{\boldsymbol{a}}{\text{Minimize}} \quad & \mathbf{1}^T \cdot \boldsymbol{e}^+ + \mathbf{1}^T \cdot \boldsymbol{e}^- \\
\text{subject to} \quad & \hat{\boldsymbol{B}} + \boldsymbol{e}^+ - \boldsymbol{e}^- = \boldsymbol{B}, \\
& \boldsymbol{e}^+ = [e_1^+, e_2^+, \cdots, e_S^+]^T \geqq \mathbf{0}, \\
& \boldsymbol{e}^- = [e_1^-, e_2^-, \cdots, e_S^-]^T \geqq \mathbf{0}, \\
& \boldsymbol{B} = [B(p_1), B(p_2), \cdots, B(p_S)]^T, \\
& \hat{\boldsymbol{B}} = [\hat{B}(p_1), \hat{B}(p_2), \cdots, \hat{B}(p_S)]^T, \\
& \hat{B}(p_i) = \boldsymbol{\theta}_i^T \cdot (\boldsymbol{a}^+ - \boldsymbol{a}^-) \\
& \quad (for\ i = 1, 2, \cdots, S), \\
& \boldsymbol{a}^+ = [a_1^+, a_2^+, \cdots, a_{20}^+]^T \geqq \mathbf{0}, \\
& \boldsymbol{a}^- = [a_1^-, a_2^-, \cdots, a_{20}^-]^T \geqq \mathbf{0}, \\
& \mathbf{1} = [1, 1, \cdots, 1]^T.
\end{aligned} \tag{4}
$$

This problem could be solved by a linear programming method such as the simplex method or the interior-point method. In this paper, we employ Barrodale's method[5] which is based on the simplex method. Also, our lossless coding scheme employs the classfication-based technique[3]: each divided block(8×8 pixels) of a image is classified to select an appropriate linear predictor based on a MMAE criterion from C different kinds of predictors, and each predictor is optimized for each class of blocks.

3. Error Distribution

We encode the error image (the difference of the original image and predictive image) using the entropy coding for lossless image coding. Thus, we need event probabilities of prediction errors. But if we encode it using actual event probabilities, the additional information will be increased, because when we decode, we need to use the same event probabilities of encode. On the other hand, if we model actual event probabilities, and encode it using model event probabilities, the additional information will be negligible, because it is only the scale parameter to set model event probabilities. So, it is important to know the shape of the error distribution.

3.1 Modeling of the error distribution

In general, the prediction error distribution is a mixture distribution consists of distributions with some different scale parameters. In order to encode effectively for this, we need to divide the prediction errors into groups by modeling. For classified groups, we obtain the high coding performance by assigning different encoders.

In this papers, we use the modeling method called context modeling.

3.2 Context modeling

A parameter U_i of a current pixel p_i is defined as follows:

$$U_i = \sum_{j=1}^{6} \frac{1}{d_{i_j}} |e_{i_j}|, \tag{5}$$

where the pixel p_{i_j} is the pixel in the local causal area given in Figure 2, and d_{i_j} is the Manhattan-distance between it and the current pixel p_i. Because the correlation between a parameter

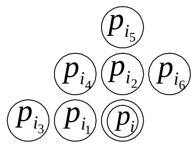


Figure 2. Region of pixels for the context modeling

U_i and a variance σ^2 of the prediction error of the current pixel p_i is very strong, we divide into groups for the prediction error using a parameter U in cotext modeling.

We use values called threshold when we divide into groups. So, we prepare thresholds as follows:

$$Th_0 \leqq Th_1 \leqq \cdots \leqq Th_{N-1},$$

where N is the number of groups. Specifically we classify the prediction errors into N groups by the parameter U. Futhermore, we optimize those thresholds in order to minimize the information value of pixels. So, we need to estimate the information value of the prediction error. The information value of the prediction error of a current pixel is defined by as follows.

Since the pixel values and the predictive values of grayscale are in the range between 0 and 255, the possible values of a prediction error are bounded between -255 and 255. Therefore, when the predictive value $\hat{B}(p_i)$ of the current pixel p_i and it's quantization group g_i are given, the conditional probability of the prediction error e_i is defined as

$$Pr_{g_i}(e_i|\hat{B}(p_i), g_i) = \frac{Pr_{g_i}(e_i|g_i)}{\sum_{x=-255}^{255} Pr_{g_i}(x|g_i)}, \tag{6}$$

where $Pr_{g_i}(x|g_i)$ is given by Probability Density Function(PDF) $f(x)$ as

$$Pr_{g_i}(x|g_i) = \int_{x-0.5}^{x+0.5} f(\xi) d\xi \tag{7}$$

with zero location parameter. Using (6), the information value of the prediction error e_i is defined as

$$J_i = -\log_2 Pr_{g_i}(e_i|\hat{B}(p_i), g_i) \tag{8}$$

Using (8), we optimize thresholds to minimize information values of the prediction error.

3.3 Shape of distribution

Error distributions of the prediction error of all quantization groups might become either a Gauss or Laplace distributions. On the other hand, the measured in a classified groups might have a different distribution.

In case of the design of predictor using MMSE, we design it so as to make large prediction errors small. As a result, very small prediction errors are disturbed. Thus, the distribution is close to a Gauss distribution and the coding performance will be improved assuming the error distribution is Gaussian. So, we expect that when we estimate the information value of the prediction error, PDF will be Gaussian.

Table 1. The Laplacian and the Gaussian function

	Laplacian	Gaussian
PDF	$\frac{1}{2b}e^{-\frac{\lvert x-\mu\rvert}{b}}$	$\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
location parameter	$\mu = \underset{i}{median}(x_i)$	$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$
scale parameter	$b = \frac{\sum_{i=1}^{n} \lvert x_i-\mu\rvert}{n}$	$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}}$

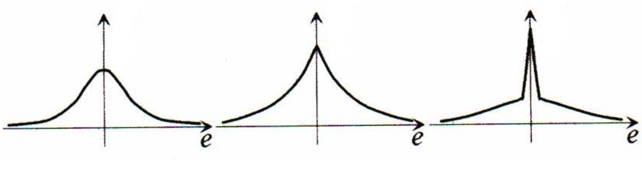
On the other hand, in case of the design of predictor using MMAE, we resolve a linear programming problem using Barrodale's method. As a result, at least k prediction errors become 0 and the predictors are designed so as to become as many as possible near 0[5]. Thus, the error distribution is close to a Laplace distribution and we can expect that the coding performance will be improved assuming the errors have a Laplace distribution for the design of predictor used MMAE. When we estimate the information value of the prediction error, PDF is Laplacian function.

From the above discussion, it is advantageous to force an actual distribution close to the Laplace distribution.

3.4 Distribution shape effect

For design of predictor using MMAE, we know at least k prediction errors become 0. In addition, k is inevitably related to the accuracy of prediction, because this is the number of pixels in the support region. If we design predictors using a small k, the number of "0" prediction errors is decreased, and the accuacy of prediction is also decresed. Thus, the error distribution will become as Figure 3-(a). On the other hand, if we design predictors using a large k, the number of "0" prediction errors is increased, and the accuacy of prediction is also increased. Thus, the error distribution will become as Figure 3-(b).

From the above discussion, we imagine that predictor order k have a distribution shaping effect and that the coding performance will be improved by shaping the error distribution as Figure 3-(c) using this effect.



(a) k : small (b) k : large (c) k : pertinence

Figure 3. Distribution shape effect.

4. Simulation

To ensure that predictor order k have a distribution shaping effect, we executed computer simulations.

When we design predictors, the number of predictor class C is one of important parameters. However, in order to prevent confusion, we fix $C = 16$ in computer simulation.

We defined a fitting degree of model distribution(laplace distribution) to actual prediction error distribution in terms of the redundancy(the difference of entropy and model entropy) and measured the redundancies, changing the value of k for several test images in Figure 4.

(a) "camera"(256×256) (b) "couple"(256×256)

Figure 4. Test images.

The entropy of prediction errors I is calculated as follows. When we denote the prediction error e's event probability $p_g(e)$ ($e = -255, -254, \cdots, 255$) in a group g, the entropy I_g of a group g is calculated by the following equation:

$$I_g = -\sum_{e=-255}^{255} p_g(e) \log_2 p_g(e)$$

Then, when we denote the group g's event probability P_g, the entropy I of all is calculated by the following equation:

$$I = \sum_{g=1}^{N} P_g I_g$$

The model entropy of prediction errors I' is calculated as follows. When we denote the prediction error e's modeled event probability $q_g(e)$ ($e = -255, -254, \cdots, 255$) in a group g, the entropy I'_g of a group g is calculated by the following equation:

$$I'_g = -\sum_{e=-255}^{255} p_g(e) \log_2 q_g(e)$$

where $p_g(e)$ is the event probability of the prediction error e. Then, when we denote the group g's event probability P_g, the model entropy I' of all is calculated by the following equation:

$$I' = \sum_{g=1}^{N} P_g I'_g$$

Because $I = I'$ if modeling is perfect, the entropy I is always smaller than the model entropy I' of prediction errors;

$$I \leqq I'.$$

Thus, the model distribution is close to an actual distribution if the redundancy($= I' - I$) is small.

Simulation results are shown in Figure 5 in next page.

In Figure 5, we can see that the redundancy is increased as k is larger for the image "camera", and it is decreased as k is larger for "couple". In "camera", the redundancy is maximum when $k = 51$, and minimum when $k = 7$. On the other hand,

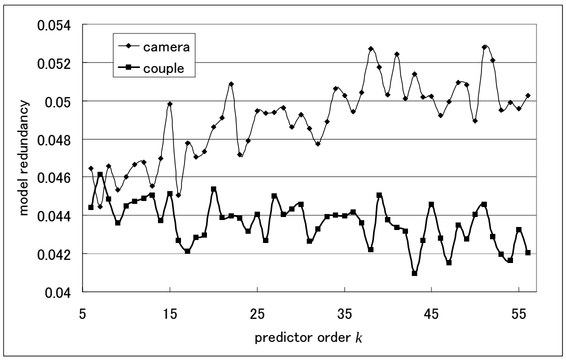


Figure 5. Simulation results.

in "couple", it is maximum when $k = 7$, and minimum when $k = 43$. In each test image, error distributions in some group when they are maximum or minimum are shown in Figure 6, 7, respectively.

In Figure 6, we can see that the model distribution is close to an actual distribution when k is small. On the other hand, in Figure 7, we can see that the model distribution is close to an actual distribution when k is large.Thus, when we encode "camera", small order predictions are enough to improve the coding performance. To the contrary, when we encode couple, we have to design predictors using large k to improve the coding performance.

5. Conclusion

From Figure 5, 6, and 7, we see that predictor order k have a distribution shaping effect, and the coding performance will be improved using the optimize k. However, the reason for rippling phenomena of Figure 5 is under consideration.

As the future work, we are due to investigate the coding simulation using this effect.

References

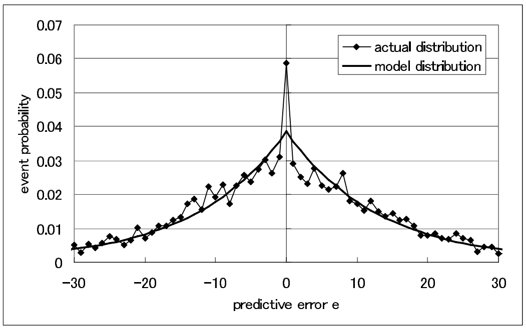
[1] B. Meyer and P. Tischer, "Glicbawls - Gray Level Image Compression by Adaptive Weighted Least Squares," Proc. Data Compression Conf. (DDC2001), p.503, Snowbird, Utah, March 2001.

[2] X. Wu, K.U. Barthel, and W. Zhang, "Piecewise 2D autoregression for predictive image coding," Proc. of Int. Conf. on Image Process. Vol. 3, 1998.

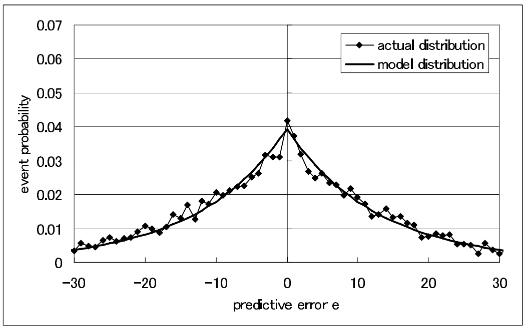
[3] F. Golchin and K. K. Paliwal, "Classified adaptive prediction and entropy coding for lossless coding of images," Proc. of 1997 IEEE Int. Conf. on Image Process. Vol. 3, pp.110-113, Oct. 1997.

[4] Y. Hashidume, Y. Morikawa and S. Maki, "Minimum Mean Absolute Error Predictors for Lossless Lmage Coding", to appear in IEICE Trans. on Information and Systems (D), vol. E91-D, no. 6.

[5] I. Barrodale, F. D. K. Roberts, "An Improved Algorithm for Discrete l1 Linear Approximation", SIAM Journal on Numerical Analysis, Vol. 10, No. 5, pp. 839-848, Oct. 1973.

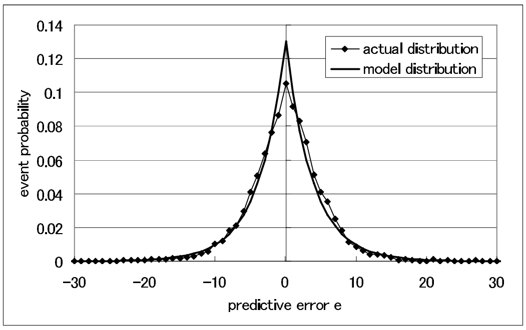


(a) $k = 51$

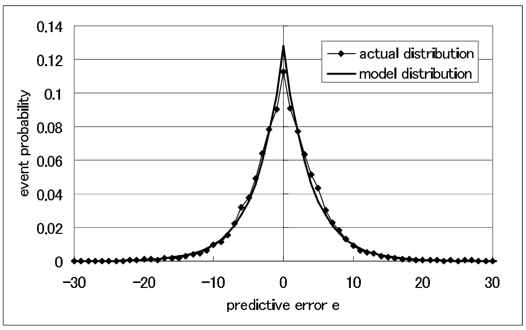


(b) $k = 7$

Figure 6. Error distribution(camera).



(a) $k = 7$



(b) $k = 43$

Figure 7. Error distribution(couple).