Memoirs of the Faculty of Engineering, Okayama University, Vol. 43, pp. 93-98, January 2009

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

Hirokazu UENO*
Graduate School of Natural Science and Technology, Okayama University
3-1-1, Tsushima-naka, Okayama, Japan

Yoshitaka MORIKAWA†
Graduate School of Natural Science and Technology, Okayama University
3-1-1, Tsushima-naka, Okayama, Japan

(Received November 28, 2008)

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

*ueno@trans.cne.okayama-u.ac.jp
†morikawa@cne.okayama-u.ac.jp

In this paper, we define the fitting degree of actual prediction error distribution and the modeling distribution(Laplace distribution) in terms of the redundancy(the difference of real 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

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 $\boldsymbol{\theta}_i = [B(p_{i_1}), B(p_{i_2}), \cdots, B(p_{i_k})]^T$ is the local causal area(support region) vector of p_i (see Fig. 1 in the next page) and $\boldsymbol{a} = [a_1, a_2, \cdots, a_k]^T$ 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:

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

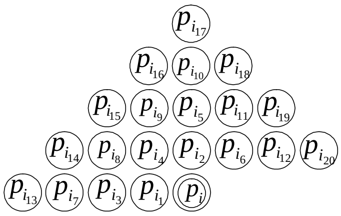


Fig. 1: Pixels of support region($k = 20$). The position of pixels in support region is defined by the Manhattan distance from the current pixel.

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:

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

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.

Table 1 lists the entropy of whole prediction errors for each the standard image[4]. The definition of the entropy is shown in section 4. From Table 1, we can see that all of the Entropy of MMAE is smaller than MMSE. Thus, the prediciotn accuracy of MMAE predictors is better than MMSE predictors.

3 ERROR DISTRIBUTION

We encode the error image (the difference of the original image and prediction image) using the entropy coding for lossless image coding. Thus, we need event probabilities

Table 1: Entropy(bits/pixel) of prediction errors only. This table indicates that the prediction accuracy of MMAE predictors is better than MMSE predictors.

Image	MMSE	MMAE
airplane	3.675	3.618
baboon	5.731	5.693
balloon	2.714	2.643
barb	4.123	3.969
barb2	4.398	4.302
camera	4.048	3.967
couple	3.428	3.386
goldhill	4.304	4.242
lena	4.353	4.295
lennagrey	3.968	3.913
peppers	4.277	4.224
Average	4.092	4.023

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 as those of encoding. On the other hand, if we model actual event probabilities, and encode errors using the model event probabilities, the additional information will be negligible, because it is only the scale parameter to determine 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 of some different scale parameters. In order to effectively tackle this problem, we divide the prediction errors into some groups by modeling. For classified groups, we may 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

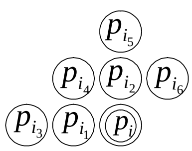


Fig. 2: Region of pixels for the context modeling.

in Fig. 2, and d_{i_j} is the Manhattan-distance between it and the current pixel p_i.

Because the correlation between parameter U_i and variance σ^2 of the prediction error of the current pixel p_i is very strong[4], we divide into some groups for the prediction error using a parameter U in cotext modeling.

We use values called threshold when we divide into some 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 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

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

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

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

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

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

Using (8), we optimize those 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 will be 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 MMSE error, Gaussian distribution will be good selection for model PDF.

On the other hand, in case of the design of predictor using MMAE, we resort to a linear programming 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 will be close to a Laplace distribution and the coding performance will be improved assuming the errors have a Laplace distribution for the design of predictor used MMAE. So, we can expect that when we estimate the information value of the MMAE error, Laplacian distribution will be good selection for model PDF. From the above discussion, it is advantageous to approximate an actual distribution by Laplace distribution in MMAE coding.

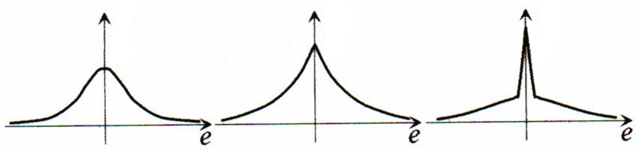
Table 2: The Laplacian and the Gaussian function.

	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	$b = \frac{\sum_{i=1}^{n} \lvert x_i-\mu\rvert}{n}$	$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}}$

3.4 Distribution Shaping 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 decreased. Thus, the error distribution will become as shown in Fig. 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 Fig. 3-(c).

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



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

Fig. 3: Distribution shaping effect of predictor order k. The coding performance may be improved by shaping the error distribution using this effect.

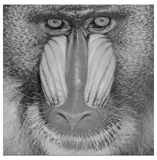
4 SIMULATION

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

When we design predictors, the number of predictor class C and the number of iterations for block classification are important parameters. However, in order to prevent confusion, we fix $C = 16$ and the number of iterations is chosen 10 in our 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 real entropy and model entropy) and measured the redundancies, changing the value of k for several test images in Fig. 4.

(a) ”airplane”(512×512)



(b) ”baboon”(512×512)

(c) ”camera”(256×256)

(d) ”couple”(256×256)

(e) ”lena”(512×512)

(f) ”lennagrey”(512×512)

(g) ”peppers”(512×512)

Fig. 4: Test images used in simulation.

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 as

$$I_g = -\sum_{e=-255}^{255} p_g(e) \log_2 p_g(e). \tag{9}$$

Then, when we denote the group g’s event probability P_g, the average entropy I is calculated as

$$I = \sum_{g=1}^{N} P_g I_g. \tag{10}$$

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 as

$$I'_g = -\sum_{e=-255}^{255} p_g(e) \log_2 q_g(e), \tag{11}$$

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 average model entropy I' is calculated as

$$I' = \sum_{g=1}^{N} P_g I'_g. \tag{12}$$

Because $I = I'$ when modeling is perfect, the real entropy I is always smaller than the model entropy I';

$$I \leqq I'.$$

Thus, the smaller the the redundancy($= I' - I$), the closer those distributions are to each other.

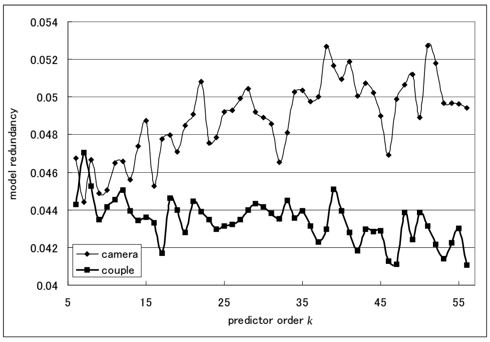
Simulation results are shown in Fig. 5 in the next page. In Fig. 5, we can see that images may be divided into two categories by the redundancy tendency. Redundancy of images in one category is increased in proportion to k like ”camera” and ”lena”, and that in the other category is decreased like ”airplane”, ”baboon”, ”couple”, ”lennagrey” and ”peppers”. Considering the vertical scale of figure, we can also see that this effect is more notable for small size images than for large size images. For some subimage of each category, error distributions in some quantization group when redundancy is maximum or minimum are shown in Fig. 6 in the next page. Fig. 6-(a) and Fig. 6-(b) are the distributions of ”camera” and ”couple”, respectively.

In Fig. 6-(a), we can see that the model distribution is close to an actual distribution when k is small. On the other hand, in Fig. 6-(b), we can see that that is close to an actual distribution when k is large. Thus, when we encode images which have the similar characteristics as ”camera”, small order k prediction selection will be preferred. To the contrary, when we encode images which have the similar characteristics as ”couple”, the large k will be preferred.

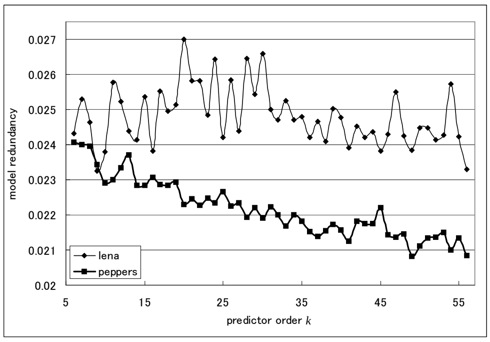
5 CONCLUSION

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

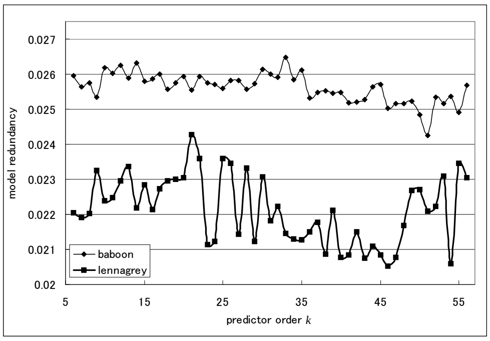
As the future work, we are due to investigate the coding simulation using this effect. Also, when we design predictors using large k to use this effect, we should devise the method to reduce side information of predictors.



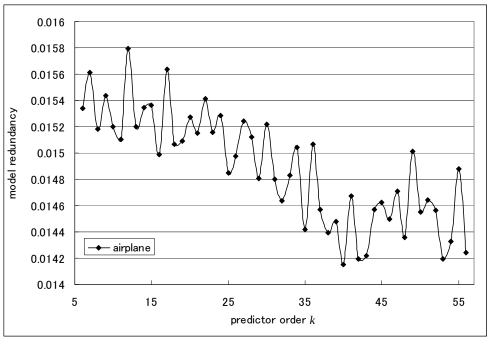
(a) ”camera” and ”couple”(256×256)



(b) ”lena” and ”peppers”(512×512)

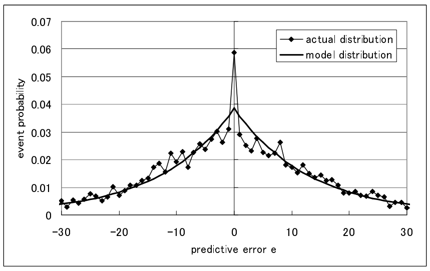


(c) ”baboon” and ”lennagrey”(512×512)

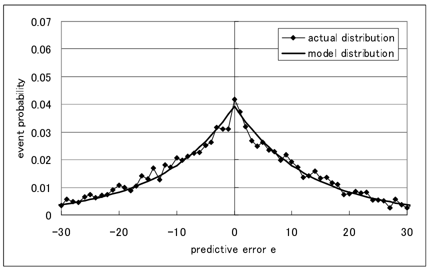


(d) ”airplane”(512×512)

Fig. 5: Simulation results. We see that images may be divided into two categories by the redundancy tendency. Redundancy of images in one category is increased in proportion to k and that in the other category is decreased.

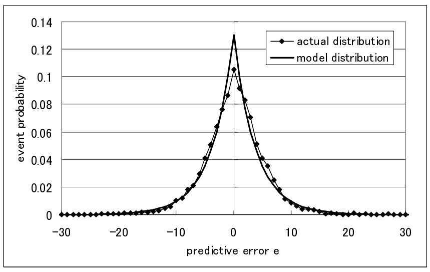


(a-1) $k = 51$

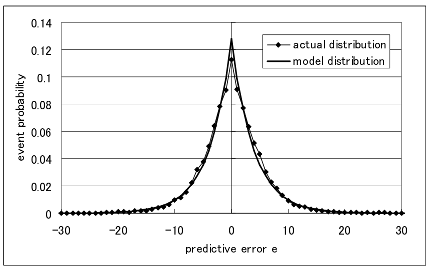


(a-2) $k = 7$

(a) ”camera”



(b-1) $k = 7$



(b-2) $k = 47$

(b) ”couple”

Fig. 6: Error distributions(”camera” and ”couple”). We can see that there are the actual distribution is close to the model distribution when k is large and it is close to the model distribution when k is small.

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, 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, pp. 901-904, 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, 1997.

[4] Y. Hashidume, Y. Morikawa and S. Maki, ”Minimum Mean Absolute Error Predictors for Lossless Image Coding”, to appear in IEICE Trans. on Inf. & Sys. (D), vol. E91-D, no. 6, pp. 1783-1792, 2008.

[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, 1973.