I-037

A Consideration on the Residual Distribution in Minimum Mean Absolute Error Prediction

Yoshihiko HASHIDUME† Yoshitaka MORIKAWA†

1. Introduction

Recent years have seen an increased level of research in lossless image compression, in addition to lossy compression. Lossless image coding is required and desired in certain applications such as medical imaging, satellite imaging and digital archiving of cultural heritages. Since the predictive coding scheme enables us to predict each pel one by one and rather precisely in aid of adaptation, many lossless coding schemes employ prediction. In prediction-based lossless image compression systems, accurate prediction and statistical error modeling play central roles and these issues are at the very heart of this research.

For lossless image coding based on prediction, the coding performance depends largely on the efficiency of predictors. In general, predictors are optimized to minimize mean square error (mse), called mmse predictors [1][2][3], but these predictors suffer from large errors at edges. In response, the authors have proposed minimum mean absolute error (mmae) predictors which are optimized to minimize mean absolute error (mae) and are less sensitive to edges [4]. [4] says that using mmae predictors the accuracy of prediction is enhanced and entropy of prediction errors is reduced.

This paper addresses the modeling of prediction errors. After designing of predictors, context-based classifications of prediction errors are conducted. The distribution of prediction errors based on mmae predictors is leptokurtic, while the one based on mmse predictors is less leptokurtic. Therefore, in this paper, we use the Laplacian and the Gaussian function models for the prediction errors based on mmae and mmse predictors, respectively, and show mmae predictors are superior to mmse predictors in terms of coding performance.

2. Minimum Mean Absolute Error Predictor

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

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

where $\boldsymbol{\theta}_i = [B(p_{i_1}), B(p_{i_2}), \cdots, B(p_{i_{14}})]^T$ is the local causal area vector of p_i (see Figure 1) and $\boldsymbol{a} = [a_1, a_2, \cdots, a_{14}]^T$ is the vector of prediction coefficients for p_i. When we denote the set of pels in coding area by $\boldsymbol{R} = \{p_i | i = 1, 2, \cdots, S\}$, a problem to design mmae predictors for $\boldsymbol{R}$ can be written as a mathematical programming problem as follows:

$$\begin{aligned} \underset{\boldsymbol{a}}{\text{Minimize}} \quad & \| \boldsymbol{e} \|_1 = \sum_{p_i \in \boldsymbol{R}} | e_i | \\ \text{subject to} \quad & \boldsymbol{e} = \boldsymbol{B} - \hat{\boldsymbol{B}}, \end{aligned} \tag{2}$$

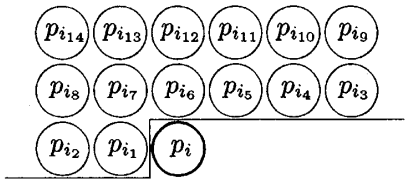


Figure 1: Region of pels for the prediction.

where $\boldsymbol{e} = [e_1, e_2, \cdots, e_S]^T$, $\boldsymbol{B} = [B(p_1), B(p_2), \cdots, B(p_S)]^T$, $\hat{\boldsymbol{B}} = [\hat{B}(p_1), \hat{B}(p_2), \cdots, \hat{B}(p_S)]^T$, and note that the parameters $\boldsymbol{a}$ to be found are included in $\hat{\boldsymbol{B}}$. 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

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

and boil down this problem to a linear programming problem. In the same way, by 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:

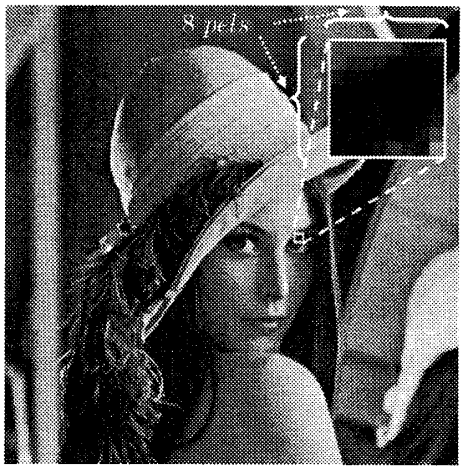
$$\begin{aligned} \underset{\boldsymbol{a}}{\text{Minimize}} \quad & \mathbf{1}^T \cdot \boldsymbol{e}^+ + \mathbf{1}^T \cdot \boldsymbol{e}^- \\ \text{subject to} \quad & \boldsymbol{B} = \hat{\boldsymbol{B}} - \boldsymbol{e}^+ + \boldsymbol{e}^-, \\ & \hat{B}(p_i) = \boldsymbol{\theta}_i^T \cdot (\boldsymbol{a}^+ - \boldsymbol{a}^-) \\ & \qquad (for\ i = 1, 2, \cdots, S), \\ & \boldsymbol{e}^\pm \geq \mathbf{0},\ \boldsymbol{a}^\pm \geq \mathbf{0}, \end{aligned} \tag{4}$$

where $\boldsymbol{e}^\pm = [e_1^\pm, e_2^\pm, \cdots, e_S^\pm]^T$, $\boldsymbol{a}^\pm = [a_1^\pm, a_2^\pm, \cdots, a_S^\pm]^T$, $\mathbf{1} = [1, 1, \cdots, 1]^T$.

This problem could be solved using linear programming methods 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, and requires minimal memory and computational time. Also, since our lossless coding scheme employs classification-based technique [3], each block (8×8 pels) of images is classified to select an appropriate linear predictor in a mmae or a mmse sense from C different kinds of predictors, and each predictor is optimized for each class of blocks.

Figure 2 (b) and (c) show the prediction errors of the sample block shown in (a) using a mmae and a mmse predictor designed for this block, respectively. Since mmae predictors are designed to deemphasize outliers, prediction accuracy of flat parts in the block is enhanced and the number of prediction errors close to 0 in the block increases than in case of mmse predictors. At the same time, prediction accuracy of edge parts in the block is dropped off and larger prediction

†Graduate School of Natural Science and Technology, Okayama University, Tsushimanaka 3-1-1, Okayama-shi, Okayama 700-8530, Japan.



(a) *sample block (lena)*

0	2	0	−10	0	3	0	7
11	25	−2	23	22	−1	−9	11
0	9	−12	15	0	−4	−8	0
0	3	0	0	3	3	6	5
0	0	−7	−4	−5	−7	−15	−12
15	0	3	−3	16	0	−10	−15
0	0	0	4	7	22	0	−19
3	0	−4	−26	−23	0	28	−33

$mae = 7.422$

$rmse = 11.244$

(b) *prediction errors based on mmae predictors*

−7	0	−2	−11	0	−6	−3	6
13	16	−7	22	21	−12	−13	12
−4	0	−10	11	−2	−5	−7	−2
−5	4	−3	−1	4	5	4	7
−4	5	−11	1	−5	−8	−17	−11
10	−1	1	−1	14	−4	−10	−15
−1	−2	0	8	5	22	−2	−17
10	−4	−6	−22	−16	−6	31	−23

$mae = 8.094$

$rmse = 10.590$

(c) *prediction errors based on mmse predictors*

Figure 2: Prediction errors based on mmae and mmse predictors. (b) and (c) show prediction errors of the sample block shown in (a) in case of mmae and mmse predictors, respectively.

Table 1: Entropy (bits/pel) of *prediction errors only*. In "mae" and "rmse" columns, mean absolute error and root mean square error of prediction errors of each image are shown, respectively, where predictors are designed for each block (8 × 8 pels) and using them prediction errors are found, and in "entropy" columns, entropy of prediction errors of each image is shown. This table indicates that using mmae predictors the accuracy of prediction is more enhanced than mmse predictors.

Image	mmae			mmse		
	mae	rmse	entropy	mae	rmse	entropy
airplane	2.364	4.660	3.519	2.556	4.297	3.768
baboon	9.014	14.963	5.274	9.678	13.937	5.714
balloon	0.953	1.612	2.469	1.018	1.501	2.515
barb	2.398	4.108	3.621	2.577	3.809	3.814
barb2	3.727	6.760	4.117	4.023	6.279	4.435
camera	3.921	9.535	3.920	4.311	8.706	4.330
couple	2.093	4.333	3.311	2.274	3.942	3.597
goldhill	3.278	5.532	4.000	3.531	5.131	4.268
lena	3.177	5.214	3.991	3.415	4.854	4.211
lennagrey	2.464	4.183	3.660	2.647	3.890	3.848
peppers	3.001	4.987	3.922	3.229	4.625	4.126
Average	3.308	5.990	3.800	3.569	5.543	4.057

errors occur. Also, Table 1 lists entropies [‡] of whole prediction errors for each image, where predictors are designed for each block and using them for each block prediction errors of each image are measured. This table implies that predictors designed in mmae sense are more effective than in mmse sense in terms of coding performance.

3. Modeling of Prediction Errors

After the prediction, context modeling of prediction errors is conducted similar to [6]. Since neighboring prediction errors are appeared to have strong correlations each other, in this paper the context function of pel p_i is defined as the weighted sum of 6 absolute prediction errors in the local causal area given in Figure 3 as follows:

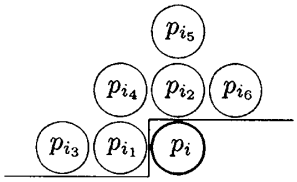


Figure 3: Region of pels for the context modeling.

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

where d_{i_j} is the Manhattan-distance of the pel p_{i_j} from the pel p_i and $e_{i_j} = \lfloor B(p_{i_j}) - \hat{B}(p_{i_j}) + 0.5 \rfloor$. According to our simulations, the weighting with the Manhattan-distance improves lossless compression rates. Using U_i and thresholds each prediction error e_i is classified to Q quantization groups (step 1), and each scale parameter (see Table 2) for each group is found (step 2). In step 1, optimal quantization group of each prediction error and thresholds can be found using dynamic programming [6]. In decoding procedure, the quantization group of each prediction error is given by U_i and the thresholds. These procedures are repeated until the likelihood $\sum_i P_{r_q}(e_i \mid \hat{B}(p_i), q)$ is maximized, where $q(= 1, 2, \cdots, Q)$ is a label of quantization groups and $P_{r_q}(\cdot)$ is the conditional probability derived from modeling probability density function (PDF) for group q. According to our simulations, in case of mmae predictors, the Laplacian function is superior to the Gaussian function as a modeling PDF for prediction errors in terms of coding performance in average. On the other hand, in case of mmse predictors, the Gaussian

[‡]Using the event probability $p(e)$ of each possible prediction error e $(= -255, -254, \ldots, 255)$ in the image, entropy I is defined as $I = -\sum_e p(e) \log p(e)$.

Table 2: The Laplacian and the Gaussian functions.

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

function is superior to the Laplacian function. So, in this paper we use the Laplacian and the Gaussian function models (given in Table 2) for the prediction errors based on mmae and mmse predictors, respectively.

4. Definition of Coding Bits

Since pel values and predicted values of grayscale imges are in the range between 0 and 255, possible values of prediction error $e_i = \lfloor B(p_i) - \hat{B}(p_i) + 0.5 \rfloor$ are bounded between -255 and 255. Therefore, when the e_i and the quantization group q of e_i are given, the conditional probability of e_i is derived as

$$P_{r_q}(e_i \mid \hat{B}(p_i), q) = \frac{P_{r_q}(e_i \mid q)}{\sum_{x=-255}^{255} P_{r_q}(x \mid q)}, \tag{6}$$

where $P_{r_q}(x \mid q)$ is given by PDF $f(x)$ as

$$P_{r_q}(x \mid q) = \int_{x-0.5}^{x+0.5} f(\xi)\, d\xi \tag{7}$$

with zero location parameter in PDF.

In this paper, as the coding bits for the prediction error e_i, the following cost function is used:

$$J_i = -\log_2 P_{r_q}(e_i \mid \hat{B}(p_i), q). \tag{8}$$

And as the coding bits for side information listed below, the entropies of them are used.

- Prediction coefficients for each class (9 bits per element of prediction coefficients including sign bit).
- Class label of prediction coefficients for each block.
- Thresholds and scale parameters for each group.

5. Optimization Procedure

As mentioned in the previous sections, in the proposed scheme, classification-based block adaptive prediction technique is employed and context-based classification of prediction errors is conducted. In this section, we describe the details of our optimization procedure in the following steps.

1. Divide the test image into blocks of size 8×8 and design a predictor for each block (“*whole pels* / (8×8)” different kinds of predictors are designed).
2. “*Whole pels* / (8×8)” predictors are narrowed down to C kinds of predictors using VQ.
3. Each block is classified to select an appropriate predictor in a mmae or a mmse sense from C different kinds of predictors.
4. Design predictors for each class of blocks.
5. Context modeling is conducted.
   (a) Optimize the quantization group of each prediction error and thresholds using dynamic programming.
   (b) Find scale parameter for each quantization group.
   (c) Repeat (a) and (b) until the likelihood $\sum_i P_{r_q}(e_i \mid \hat{B}(p_i), q)$ is maximized.
6. Stop if mae or mse is minimized, else go to 3..

6. Experimental Results

In order to evaluate the efficiency of mmae predictors, we compare them with mmse predictors. The experimental conditions are equal to each other all but predictors and the modeling function. Table 3 and 4 list compression results tested for several 8-bit continuous-tone monochrome images. In our simulation, the number of Q is fixed to 16 and C is set to $16, 32, 64$. In Table 3, as coding bits of prediction errors the entropy of them is used. The results show that accurate prediction with mmae predictors improves coding performance. In Table 4, as coding bits of prediction error e_i the cost function J_i given by Eq.(8) is used. The results show that mmae predictors are superior to mmse predictors in terms of coding performance, in each case of C, including error modeling. Especially, bit rates for ‘camera’ and ‘couple’ are reduced very much. On the other hand, bit rates for ‘balloon’ are increased. This is because the Gaussian function is more suitable than the Laplacian as the modeling PDF for this image. This problem can be taken care of by employing the generalized Gaussian function [6] or the generalized Laplacian function [7] as the modeling function.

Table 3: Entropy (bits/pel) of *whole information* needed to encode several 8-bit continuous-tone monochrome images in case of mmae and mmse predictors, *where as coding bits of prediction errors the entropy of them is used.* This table indicates that accurate prediction improves coding performance.

Image	mmae			mmse		
	16	32	64	16	32	64
airplane	3.679	3.687	3.702	3.693	3.696	3.716
baboon	5.747	5.754	5.776	5.751	5.757	5.774
balloon	2.722	2.720	2.729	2.723	2.719	2.726
barb	4.004	3.973	3.957	4.057	3.982	3.966
barb2	4.346	4.339	4.346	4.404	4.389	4.370
camera	4.049	4.077	4.137	4.121	4.144	4.219
couple	3.493	3.517	3.583	3.536	3.561	3.627
goldhill	4.293	4.294	4.307	4.310	4.303	4.311
lena	4.345	4.346	4.362	4.356	4.350	4.365
lennagrey	3.957	3.959	3.972	3.961	3.966	3.980
peppers	4.278	4.284	4.296	4.279	4.283	4.298
Average	4.083	4.086	4.106	4.108	4.105	4.123

Table 4: Bit rates (bits/pel) of *whole information* needed to encode several 8-bit continuous-tone monochrome images in case of mmae and mmse predictors. The number (16, 32, 64) written in the second row indicates the number of classes of predictors.

Image	mmae			mmse		
	16	32	64	16	32	64
airplane	3.706	3.710	3.722	3.736	3.734	3.750
baboon	5.778	5.781	5.800	5.788	5.793	5.806
balloon	2.772	2.767	2.772	2.740	2.733	2.739
barb	4.045	4.012	3.993	4.076	4.000	3.981
barb2	4.379	4.370	4.372	4.442	4.423	4.399
camera	4.108	4.131	4.189	4.226	4.236	4.293
couple	3.527	3.546	3.612	3.611	3.630	3.684
goldhill	4.321	4.317	4.327	4.346	4.335	4.340
lena	4.377	4.375	4.387	4.379	4.373	4.385
lennagrey	3.988	3.988	3.997	3.986	3.990	4.001
peppers	4.307	4.309	4.318	4.309	4.309	4.323
Average	4.119	4.119	4.135	4.149	4.141	4.155

7. Computational Complexity

In the normal simplex method, a lot of discharge calculations are needed. So, computational time to solve linear programming problems depends largely on the order k of predictors and the number S of pels in each prediction class. In addition, a simplex tableau which is an array of dimentions $(S+1) \times 2(k+S+1)$ is required. In our case, k is fixed to 14, but in the iteration procedure noted Sec. 5. S happens to be over 10000. Therefore, this method is not realistic with respect to computational time and memory. In response, the Barrodale's method [5] requires small size of simplex tableau, an array of dimentions $(S+2) \times (k+2)$, and less discharge calculations. On the images whose size is 512×512 pels, our scheme takes about $10-15$ minutes with 3.60GHz Pentium(R) 4 processor.

8. Conclusion

We considered the residual distribution in minimum mean absolute error prediction. Experimental results demonstrate that accurate prediction improves coding performance and mmae predictors are effective in lossless image coding. In this paper, we employed simple PDFs such as the Laplacian and the Gaussian function as modeling functions. Since prediction errors have some kinds of statistical structures, these functions might not be sufficient for modeling. The generalized Laplacian function, the generalized Gaussian function, or other mixtured models enable us to model prediction errors more precisely and more reduction of the bit rates is expected.

Also, as future works, improvement of predictors, coding of side information, reduction of computational time, implementation of arithmetic coding, and etc. are desired.

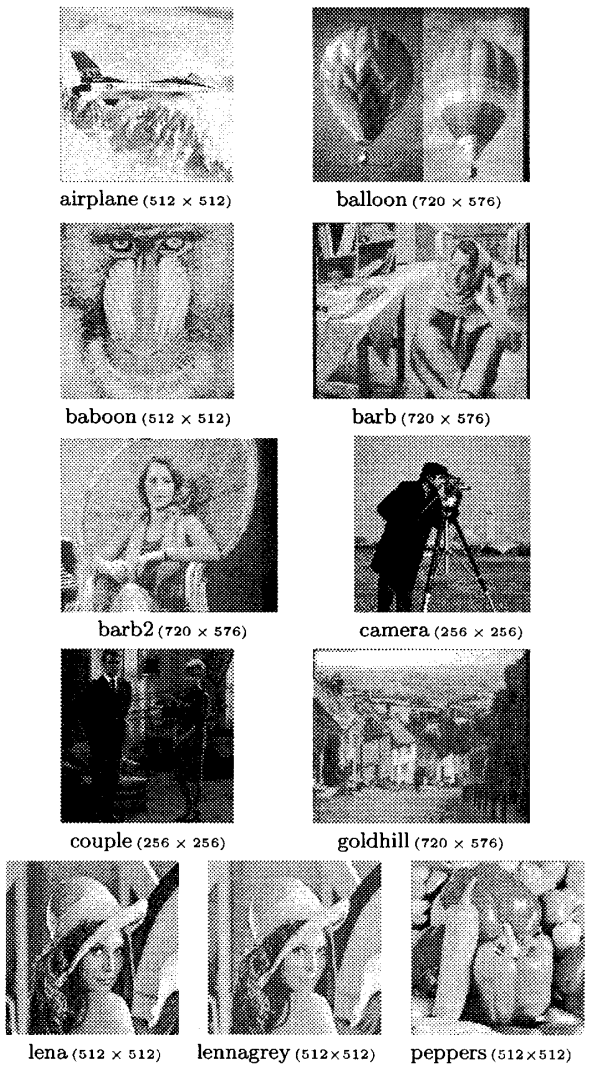


Figure 4: Test images used in simulations.

References

[1] B. Meyer and P. Tischer, "Glicbawls – Grey Level Image Compression by Adaptive Weighted Least Squares," Proc. Data Compression Conf. (DCC 2001), 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, S. Maki, Y. Morikawa, N. Yamane, "A Consideration on Adaptive Prediction Filters Minimizing Mean Absolute Errors", Proc. of 57th Rengou-taikai, Chugoku-branch on Institues for Electrical, Information and Communication Engineers, pp.139-140, Oct. 2006.

[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.

[6] I. Matsuda, N. Ozaki, Y. Umezu and S. Itoh, "Lossless Coding Using Variable Block-size Adaptive Prediction Optimized for Each Image," Proc. of 13th European Signal Process. Conf. (EUSIPCO 2005), WedAmPO3, Sep. 2005.

[7] A. Nakamura, "Acoustic Modeling for Speech Recognition Based on a Generalized Laplacian Mixture Distribution," IEICE Trans. Information and Systems(D-II), Vol. J83-D-II, No. 11, pp. 2118-2127, Nov. 2000.