LOSSLESS CODING USING VARIABLE BLOCK-SIZE ADAPTIVE PREDICTION OPTIMIZED FOR EACH IMAGE

Ichiro Matsuda, Nau Ozaki, Yuji Umezu and Susumu Itoh

Department of Electrical Engineering, Faculty of Science and Technology,
Science University of Tokyo
2641 Yamazaki, Noda-shi, Chiba 278-8510, JAPAN
Tel: +81 4 7124 1501 (ex.3722); Fax: +81 4 7124 9367
E-mail: matsuda@itohws01.ee.noda.sut.ac.jp

ABSTRACT

This paper proposes an efficient lossless coding scheme for still images. The scheme employs a block-adaptive prediction technique to remove spatial redundancy in a given image. The resulting prediction errors are encoded using context-adaptive arithmetic coding. Several coding parameters, which must be sent to a decoder as side information, are iteratively optimized for each image so that the number of coding bits including the side information can have a minimum. Moreover, quadtree-based variable block-size partitioning is introduced into the above adaptive prediction technique. Experimental results show that the proposed coding scheme attains the best coding performance among the current state-of-the-art lossless coding schemes.

1. INTRODUCTION

Linear prediction is often used for lossless image coding as a simple and effective tool to remove spatial redundancy in a given image. To attain accurate prediction for nonstationary images, several techniques which adapt prediction coefficients according to local characteristics of image signals are proposed. Basically, there are two approaches for such adaptive prediction techniques, namely, backward-adaptive and forward-adaptive prediction. Since the backward-adaptive prediction exploits information in a local causal area, it can adapt prediction coefficients pel-by-pel without any side information [1, 2]. However, this approach usually increases complexity of not only encoding but also decoding because the identical operation for the adaptation must be carried out at both encoder and decoder sides. On the other hand, the forward-adaptive prediction generally gives stable performance owing to use of information unavailable at the decoder [3, 4, 5]. In addition, it can be fast in decoding because most of expensive computation needed for the adaptation is performed only at the encoder side.

From this point of view, we previously proposed a novel lossless coding scheme based on the forward-adaptive prediction [6]. The scheme takes advantage of a block-adaptive prediction technique where a set of linear predictors are designed for each image and an appropriate one is selected from the set block-by-block. Unlike the conventional techniques [4, 5] which utilize predictors designed in a minimum mean square error (MMSE) sense, our scheme iteratively optimizes the predictors so that a cost function corresponding to the number of coding bits required for the prediction errors can have a minimum. As a result, the obtained predictors provide better coding performance than the conventional MMSE-based predictors. Nevertheless, the coding scheme is not necessarily optimum because the cost function does not reflect the amount of side information which is needed to carry out the adaptive prediction at the decoder side.

In this paper, we introduce a new cost function which is defined as the sum of coding bits for the prediction errors and the side information. Moreover, a variable block-size adaptive prediction method using the above cost function is developed to enable global optimization of the coding rate including the side information on predictor selection. The effectiveness of the proposed coding scheme is confirmed through comparison with current state-of-the-art lossless coding schemes.

2. THE PROPOSED CODING SCHEME

Our lossless coding scheme employs the block-adaptive prediction technique [4] which partitions an image into square blocks and classifies them into several classes. Each class has an individual predictor which is optimized for blocks of the same class. When the current pel $\boldsymbol{p}_0$ is in the block belonging to the m-th class ($m = 1, 2, \ldots, M$), a predicted value $\hat{s}(\boldsymbol{p}_0)$ is calculated by the following equation:

$$\hat{s}(\boldsymbol{p}_0) = \sum_{k=1}^{K} a_m(k) \cdot s(\boldsymbol{p}_k), \qquad (1)$$

where $a_m(k)$s are prediction coefficients of the m-th predictor, $\boldsymbol{p}_k$s are pels used for the prediction ($k = 1, 2, \ldots, K$) and $s(\boldsymbol{p}_k)$ represents a value of image signals at the pel $\boldsymbol{p}_k$. Figure 1 shows disposition of the pels used in Eq. (1).

After the prediction, context modeling for adaptive arithmetic coding of the prediction error $e = s(\boldsymbol{p}_0) - \hat{s}(\boldsymbol{p}_0)$ is conducted. This context modeling is based on non-linear

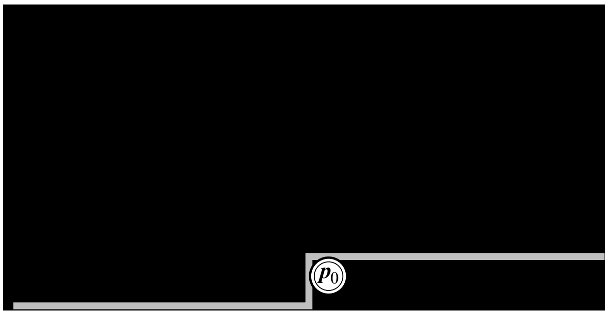

Figure 1: Disposition of pels for the prediction ($K = 30$).

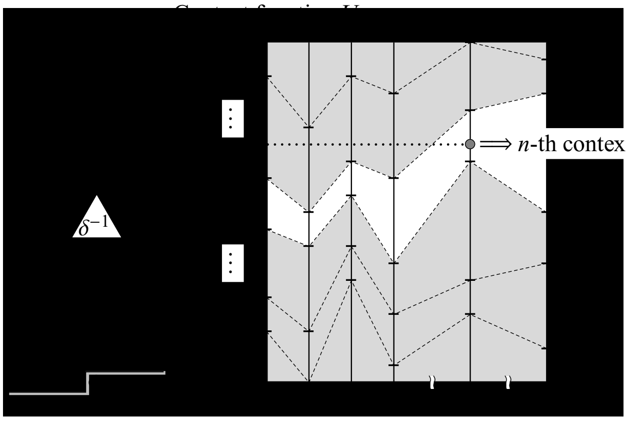

Figure 2: Context modeling for adaptive arithmetic coding.

quantization of a context function which captures statistical property of the prediction errors in a local causal area [7]. In this paper, the context function is defined as the weighted sum of absolute prediction errors at already encoded twelve pels.

$$U = \sum_{k=1}^{12} \frac{1}{\delta_k} \cdot \left| s(\boldsymbol{p}_k) - \hat{s}(\boldsymbol{p}_k) \right|, \qquad (2)$$

where δ_k is the Euclidean distance of the pel $\boldsymbol{p}_k$ from the current pel $\boldsymbol{p}_0$. Each quantization level of U corresponds to one of sixteen contexts ($n = 1, 2, \ldots, 16$) as shown in Figure 2 and thresholds $\{Th_m(1), Th_m(2), \ldots, Th_m(15)\}$ used in this quantization are optimized for each class (m) as described later. For each context, we then consider that a conditional probability density function (PDF) of the prediction error e can be modeled by the generalized Gaussian function [6]:

$$P(e\,|\,n) = \frac{c_n \cdot \eta(c_n, \sigma_n)}{2\Gamma(1/c_n)} \cdot \exp\left\{ -\left| \eta(c_n, \sigma_n) \cdot e \right|^{c_n} \right\},$$
$$\eta(c_n, \sigma_n) = \frac{1}{\sigma_n} \sqrt{\frac{\Gamma(3/c_n)}{\Gamma(1/c_n)}}, \qquad (3)$$

where $\Gamma(\cdot)$ is the gamma function, σ_n is standard deviation of e and c_n is a shape parameter which controls sharpness of the PDF. In this scheme, a fixed value of σ_n is given for each context [6]. Since 8-bit grayscale images contain integer values from 0 to 255, possible values of the prediction error e for a given $\hat{s}(\boldsymbol{p}_0)$ are also limited to the following 256 values:

$$e \in \{ s - \hat{s}(\boldsymbol{p}_0) \,|\, s = 0, 1, 2, \ldots, 255 \}. \qquad (4)$$

Therefor, a conditional probability of occurrence of each possible value of e, when both the context n and the predicted value $\hat{s}(\boldsymbol{p}_0)$ are known, is derived from the above PDF.

$$\Pr(e\,|\,\hat{s}(\boldsymbol{p}_0), n) = \frac{\Pr(e\,|\,n)}{\sum_{s=0}^{255} \Pr(s - \hat{s}(\boldsymbol{p}_0)\,|\,n)}, \qquad (5)$$

$$\Pr(e\,|\,n) = \int_{-h_s/2}^{h_s/2} P(e + \varepsilon\,|\,n)\, \mathrm{d}\varepsilon. \qquad (6)$$

Actually, the predicted value $\hat{s}(\boldsymbol{p}_0)$ is explicitly rounded to the nearest multiples of h_s to avoid accumulation of unexpected rounding errors. Hence the value of h_s is used as an interval of integration of the PDF in Eq. (6). Adaptive arithmetic coding of the actual value of e is carried out according to the conditional probabilities calculated by Eqs. (5) and (6). Note that the numerator of Eq. (5) corresponds to the area shown in dark gray and the denominator is the sum of the shaded areas in Figure 3. It illustrates that a small value of h_s allows accurate modeling of the probabilities in consideration of a fractional part of the predicted value ($\hat{s}(\boldsymbol{p}_0) - \lfloor \hat{s}(\boldsymbol{p}_0) \rfloor$). In this case, by storing all of the probabilities in a look-up table at a sampling rate of $1/h_s$, we can considerably reduce the computation required for the adaptive arithmetic coding. Our experiments show that a value of $h_s = 1/8$ is a reasonable choice in terms of coding efficiency and memory consumption for the look-up table.

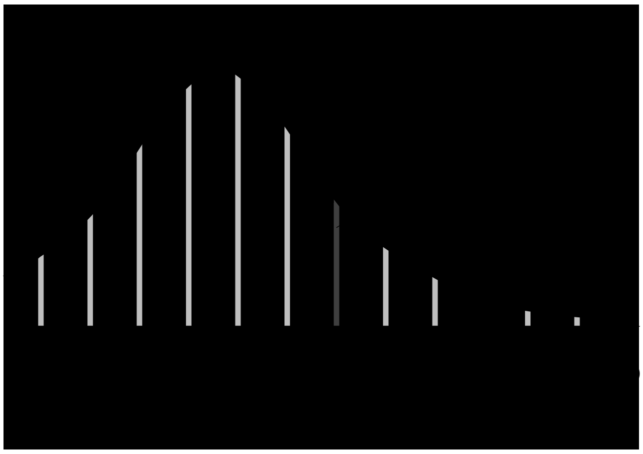

Figure 3: Conditional probability of occurrence of the prediction error e.

3. OPTIMIZATION OF CODING PARAMETERS

In the proposed lossless coding scheme, parameters listed below must be sent to a decoder as side information.

- Prediction coefficients $\{a_m(k)\,|\,k = 1, \ldots, K\}$ for each class.
- Thresholds $\{Th_m(1), Th_m(2), \ldots, Th_m(15)\}$ for each class.
- Class label m for each block.
- Shape parameter c_n for each context.

Values of these parameters are iteratively optimized for each image so that the following cost function can have a minimum.

$$J = -\sum_{\boldsymbol{p}_0} \log_2 \Pr(e\,|\,\hat{s}(\boldsymbol{p}_0), n) + B_{side}. \qquad (7)$$

The first term of the cost function represents the number of coding bits required for the adaptive arithmetic coding of the prediction errors. On the other hand, the second term (B_{side}) indicates the amount of side information on the above coding parameters. Since these parameters are encoded by using arithmetic code and proper probability models, a value of B_{side} continually changes in the optimization process. In our previous work [6], the cost function did not include the term of B_{side} and, as a matter of course, the optimization process could not minimize the overall coding rate. It is expected that use of the new cost function given by Eq. (7) yields a better trade-off between the number of coding bits for the prediction errors and that for the side information. Concrete procedures for the optimization are as follows.

(1) Classify every block into one of M classes according to variance of image intensity within the block, and design initial predictors for individual classes using a fast method developed on the assumption of the Gaussian PDF [8].
(2) Choose two prediction coefficients $a_m(i)$ and $a_m(j)$ randomly, and carry out partial optimization by varying values of them gradually. Repeat the partial optimization a certain number of times in each class.
(3) Optimize the thresholds $\{Th_m(1), Th_m(2), \ldots, Th_m(15)\}$ in each class by using the dynamic programming technique.
(4) Select the optimum value of the shape parameter c_n from sixteen values $(0.2, 0.4, 0.6, \ldots, 3.2)$ in each context.
(5) Re-classify all the blocks by selecting the optimum predictor, or the optimum class in each block.
(6) Repeat the procedures (2), (3), (4) and (5) until all the coding parameters converge.

4. VARIABLE BLOCK-SIZE ADAPTIVE PREDICTION

In the block-adaptive prediction technique, adoption of smaller block-size obviously improves accuracy of the adaptive prediction, while it increases the amount of side information on class labels (m). Past researches [4, 5, 6] found that the block-size of 8×8 pels was reasonable in terms of the average coding performance. However, the best block-size generally depends on local characteristics of image signals and adaptation of the block-size must improve the coding performance. Thereupon, we introduce quadtree-based variable block-size partitioning [9] into the block-adaptive prediction. A quadtree is built by recursive partitioning of a square block into four sub-blocks. Each node of the tree, except in the lowest level, has a flag which indicates whether the corresponding block is further partitioned '1' or not '0'. Accordingly, an arbitrary partitioning pattern of blocks based on the quadtree structure can be represented by a series of such flags as shown in Figure 4. In this paper, five-level partitioning with a maximum block-size of 32×32 pels is performed and the best combination of both block-sizes and class labels which minimizes the cost function J is determined. The number of bits required for the above quadtree flags is also added to B_{side} when the cost function is computed. This quadtree decision process is iteratively performed in the optimization procedures described in the previous section. To be exact, it is carried out instead of the procedure (5) for every iteration.

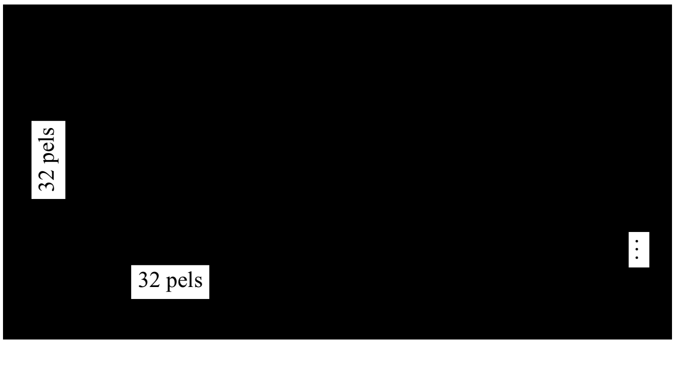

Figure 4: Example of quadtree-based variable block-size partitioning.

5. EXPERIMENTAL RESULTS

To evaluate practical coding performance of the proposed scheme, we have developed a software-based codec using C language and tested it for several monochrome images, all of which are digitized in 8-bit grayscale. Table 1 lists coding rates of our coding schemes. 'VBS & new-cost' means the proposed scheme which employs both the variable block-size adaptive prediction and the new cost function defined by Eq. (7). 'FBS & new-cost' employs the same cost function but block-size is fixed to 8×8 pels. And 'FBS' is our previous scheme [6] where the amount of side information (B_{side}) is omitted in the cost function and the fixed block-size of 8×8 pels is used for the adaptive prediction. The number of predictors (M) and prediction order (K) used in the experiments are shown in the table as well. These values have been preliminarily determined for each image size so that they give better results in average. The table shows that coding rates of 'VBS & new-cost' are 0.004–0.070 and 0.029–0.139 bits/pel lower than those of 'FBS & new-cost' and 'FBS', respectively. Table 2 reports

Table 1: Comparison of coding rates (bits/pel).

Image	Size, M, K	VBS & new-cost	FBS & new-cost	FBS [6]
Camera	256×256, $M=20$, $K=30$	3.949	3.989	4.013
Couple		3.388	3.403	3.429
Noisesquare		5.270	5.288	5.312
Airplane	512×512, $M=41$, $K=42$	3.591	3.600	3.622
Baboon		5.663	5.667	5.698
Lena		4.280	4.288	4.314
Lennagrey		3.889	3.900	3.924
Peppers		4.199	4.207	4.233
Shapes		0.685	0.755	0.824
Balloon	720×576, $M=56$, $K=72$	2.579	2.584	2.608
Barb		3.815	3.827	3.856
Barb2		4.216	4.224	4.258
Goldhill		4.207	4.215	4.245
Average		*3.826*	*3.842*	*3.872*

Table 2: Detailed coding results for the 'Camera' (bits).

Item	VBS & new-cost	FBS & new-cost	FBS [6]
Coefficients	2160	2304	3464
Thresholds	808	848	1264
Class labels	4304	2696	3272
Block-size	960	–	–
Shape parameters	64	64	64
Prediction errors	250424	255384	254832
Others	110	110	110
Total	*258830*	*261406*	*263006*

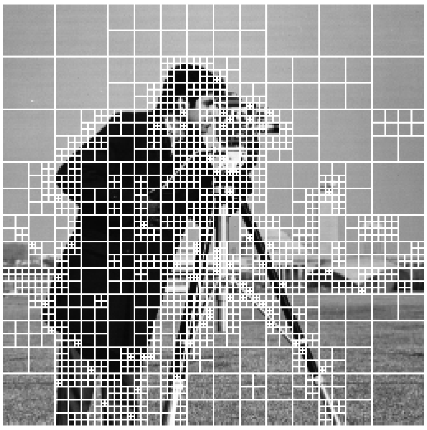

Figure 5: Result of variable block-size partitioning (Camera).

Table 3: Performance comparison with the state-of-the-art lossless coding schemes (bits/pel).

Image	VBS & new-cost	BMF	TMW	Glicbawls	JPEG-LS	JPEG 2000
Camera	3.949	4.060	4.098	4.208	4.314	4.535
Couple	3.388	3.448	3.446	3.543	3.699	3.915
Noisesquare	5.270	5.298	5.542	5.415	5.683	5.634
Airplane	3.591	3.602	3.601	3.668	3.817	4.013
Baboon	5.663	5.714	5.738	5.666	6.037	6.107
Lena	4.280	4.317	4.300	4.295	4.607	4.684
Lennagrey	3.889	3.929	3.908	3.901	4.238	4.303
Peppers	4.199	4.241	4.251	4.246	4.513	4.629
Shapes	0.685	0.730	0.740	2.291	1.214	1.926
Balloon	2.579	2.649	2.649	2.640	2.904	3.031
Barb	3.815	3.959	4.084	3.916	4.691	4.600
Barb2	4.216	4.276	4.378	4.318	4.686	4.789
Goldhill	4.207	4.238	4.266	4.276	4.477	4.603
Average	*3.826*	*3.882*	*3.923*	*4.030*	*4.222*	*4.367*

breakdowns of the number of coding bits for the ‘Camera’ image. It indicates that use of the new cost function can remarkably reduce the amount of side information, while it slightly increases the number of coding bits required for the prediction errors. In addition, introduction of variable block-size adaptive prediction can provide a better trade-off between the number of coding bits required for the prediction errors and the side information. From the result of variable block-size partitioning shown in Figure 5, we can confirm that appropriate block-sizes according to complexity of local image structures are obtained.

Table 3 also lists coding rates of the proposed scheme ‘VBS & new-cost’ together with those of the state-of-the-art lossless coding schemes: BMF [10], TMW [3], Glicbawls [2], JPEG-LS [11] and JPEG 2000 [12]. It is demonstrated that the proposed scheme attains the best coding performance for all tested images and its coding rates are 6–44 % lower than those of the JPEG-LS standard.

6. CONCLUSIONS

We have proposed an efficient lossless image coding scheme using variable block-size adaptive prediction. In order to improve coding efficiency, several coding parameters including the block-size are optimized for each image so that a cost function which is defined as the sum of coding bits for prediction errors and side information can have a minimum. Experimental results demonstrate that the proposed coding scheme outperforms the other state-of-the-art schemes in terms of coding efficiency.

Due to the nature of forward-adaptive prediction, the proposed scheme requires relatively large amount of computation at the encoder side but is practically fast in decoding process. When the size of images is 512×512 pels, for example, our codec takes 10–25 minutes for encoding and at most 0.2 seconds for decoding on a computer with the 3.06 GHz Xeon processor. This feature is useful for client/server applications where offline encoding at the server side is allowable.

REFERENCES

[1] X. Wu, K. U. Barthel and W. Zhang, “Piecewise 2D Autoregression for Predictive Image Coding,” Proc. of 1998 IEEE Intl. Conf. on Image Processing (ICIP ’98), Vol.III, pp.901–904, Oct. 1998.

[2] B. Meyer and P. Tischer, “Glicbawls - Grey Level Image Compression by Adaptive Weighted Least Squares,” Proc. of 2001 Data Compression Conf. (DCC 2001), p.503, Mar. 2001.

[3] B. Meyer and P. Tischer, “TMW – a New Method for Lossless Image Compression,” Proc. of 1997 Picture Coding Symposium (PCS ’97), pp.533–538, Sep. 1997.

[4] F. Golchin and K. K. Paliwal, “Classified Adaptive Prediction and Entropy Coding for Lossless Coding of Images,” Proc. of 1997 IEEE Intl. Conf. on Image Processing (ICIP ’97), Vol.III, pp.110–113, Oct. 1997.

[5] B. Aiazzi, L. Alparone and S. Baronti, “Near-Lossless Image Compression by Relaxation-Labelled Prediction,” Signal Processing, vol.82, no.10, pp.1619–1631, Oct. 2002.

[6] I. Matsuda, N. Shirai and S. Itoh, “Lossless Codling Using Predictors and Arithmetic Code Optimized for Each Image,” Proc. of Intl. Workshop VLBV03, LNCS, Vol.2849, pp.199–207, Sep. 2003.

[7] B. Aiazzi, L. Alparone and S. Baronti, “Context Modeling for Near-Lossless Image Coding,” IEEE Signal Processing Letters, Vol.9, No.3, pp.77–80, Mar. 2002.

[8] I. Matsuda, H. Mori and S. Itoh, “Lossless Coding of Still Images Using Minimum-Rate Predictors,” Proc. of 2000 IEEE Intl. Conf. on Image Processing (ICIP-2000), Vol.I, pp.132–135, Sep. 2000.

[9] G. J. Sullivan and R. L. Baker, “Efficient Quadtree Coding of Images and Video,” Proc. of 1991 IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-91), Vol.4, pp.2661–2664, Apr. 1991.

[10] http://compression.ru/ds/

[11] ISO/IEC, ISO/IEC 14495-1:1999, “Information Technology – Lossless and Near-lossless Compression of Continuous-Tone Still Images: Baseline,” Dec. 1999.

[12] ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FCD 15444-1, “Information Technology – JPEG 2000 Image Coding System,” Mar. 2000.