Least Squares-Based Lossless Image Coding with Edge-look-ahead

Lih-Jen Kau, *Student Member, IEEE* and Yuan-Pei Lin, *Senior Member, IEEE*
Dept. Elec. and Control Engr., National Chiao Tung Univ.
1001 Ta Hsueh Road, Hsinchu, Taiwan, R.O.C.
Tel/Fax: 886-3-8236990
Email: u8912801@cc.nctu.edu.tw, ypl@cc.nctu.edu.tw

***Abstract*— In predictive image coding, the least squares (LS)-based adaptive predictor is noted as an efficient method to improve prediction result around edges. However pixel-by-pixel optimization of the predictor coefficients leads to a high coding complexity. To reduce computational complexity, we activate the LS optimization process only when the coding pixel is around an edge or when the prediction error is large. We propose a simple yet effective edge detector using only causal pixels. The system can look ahead to determine if the coding pixel is around an edge and initiate the LS adaptation to prevent the occurrence of a large prediction error. Our experiments show that the proposed approach can achieve a noticeable reduction in complexity with only a minor degradation in the prediction results.**

I. Introduction

Many of the recent advances in lossless image coding are based on predictive coding with context modeling [1]-[7]. Moreover, the image model is assumed to be stationary during prediction. However, this rarely happens in the real world and large prediction errors can take place especially when the coding pixel is around edges. Recently, linear predictors adapted by least squares (LS) optimization have been proposed as an efficient approach to accommodate varying statistics of coding images [2][3]. Among which, the EDP [2] predictor pointed out that the superiority of LS optimization is in its edge-directed property. For complexity consideration, performing the LS adaptation process in a pixel-by-pixel manner is regarded as prohibitive. Therefore, the EDP [2] proposed initiating the LS optimization process only when the prediction error is beyond a preselected threshold such that the computational complexity can be reduced. The EDP [2] has made a noticeable improvement over the state-of-the-art lossless coder CALIC [4].

As large prediction errors usually take place in pixels around an edge, the prediction result can be improved if we can foresee the existence of an edge. Therefore, we propose an adaptive predictor with edge-look-ahead which can fully exploit the edge-directed characteristic of the LS-based adaptation process. To do this, we propose a simple and efficient edge detector using only causal pixels, i.e., pixels that have already been coded. With the proposed edge detector, the predictor can determine if the coding pixel is around an edge and initiate the LS adaptation process beforehand to prevent the occurrence of a large prediction error. We will see that the proposed edge detector, though very simple, can pick out the edges successfully in the experiments. Our experiments also show that a very good tradeoff between the computational complexity and the prediction result can be obtained.

The rest of the paper is organized as follows. Section II introduces the proposed "*Edge Detector*". The LS-based adaptive predictor is given in section III. Experimental results of the proposed method and comparisons to existing predictors and coders are given in section IV. A conclusion is given in section V.

II. Edge Detector

To determine whether the coding pixel is around an edge, we propose a very simple algorithm that uses only causal pixels. It should be noted that conventional edge detectors, e.g., "Sobel" operator, can not be applied here because they use non-causal pixels.

We observe that the variance of an area that contains an edge is usually large. Furthermore, the histogram of such an area tends to have two peaks, one on each side of the mean value. We will use these two observations to determine the existence of an edge. We define the *texture context* κ of a coding pixel as the collection of the four nearest causal pixels $x_n(1), \cdots, x_n(4)$ in Fig. 1.

The mean $\bar{x}$ and variance σ^2 of the *texture context* are calculated. Moreover, the four pixels can be divided into two groups, the pixels with gray levels higher than $\bar{x}$ in one group κ_h and the rest in another κ_l. We also compute the variance σ_h^2, σ_l^2 of those pixels in κ_h and κ_l respectively.

We determine whether the coding pixel is around an edge if the following two conditions are both satisfied,

$$\sigma^2 \geq \gamma_1, \quad \text{and} \quad \frac{\sigma^2}{0.01 + \sigma_h^2 + \sigma_l^2} \geq \gamma_2, \tag{1}$$

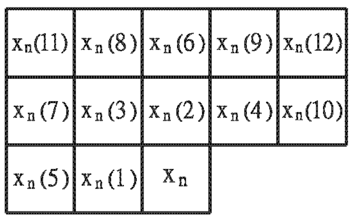


Fig. 1. The ordering of pixels for prediction inputs.

where 0.01 is added so that the denominator of (1) does not become 0 when σ_h^2 and σ_l^2 are both zero. We have found through experiments that $\gamma_1 = 100$ and $\gamma_2 = 10$ work very well and these values will be used throughout the paper.

III. The LS-based Adaptive Prediction

In this paper, the predicted value of the coding pixel is a linear combination of its causal neighbors. The corresponding inputs for different prediction orders are shown in Fig. 1 where the ordering of pixels is based on the distance to the pixel to be encoded. Therefore, the predicted value $\hat{x}_n$ of x_n is given by

$$\hat{x}_n = \sum_{k=1}^{N} a(k) x_n(k), \tag{2}$$

where N is the prediction order, $x_n(k)$ is the kth nearest neighbor of x_n and $a(k)$ is the corresponding predictor coefficient.

To adapt the predictor to the varying statistics around the coding pixel, the LS-based adaption process is activated whenever the two conditions in (1) are satisfied or when the prediction error is greater than a predefined threshold. Suppose we have M pixels in the training area, our objective is to find a least-square solution for the system

$$\mathbf{Pa} = \mathbf{y}, \tag{3}$$

where

$$\mathbf{P} = \begin{bmatrix} x_{n-1}(1) & x_{n-1}(2) & \dots & x_{n-1}(N) \\ x_{n-2}(1) & x_{n-2}(2) & \dots & x_{n-2}(N) \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-M}(1) & x_{n-M}(2) & \dots & x_{n-M}(N) \end{bmatrix}$$

is an $M*N$ matrix with its rows consisting of the N neighbors of the M training pixels, $\mathbf{a} = [a(1), a(2), \cdots, a(N)]^T$ is the Nth order predictor coefficient vector to be determined and $\mathbf{y} = [x_{n-1}, x_{n-2}, \cdots, x_{n-M}]^T$ is the M-dimensional vector consisting of the M training pixels.

To minimize the square errors $\|\mathbf{y} - \mathbf{Pa}\|_2^2$ for (3), the normal equations below provides the key for the solution,

$$\mathbf{P}^T\mathbf{Pa} = \mathbf{P}^T\mathbf{y} \tag{4}$$

If we define $\mathbf{B} = \mathbf{P}^T\mathbf{P}$ and $\mathbf{c} = \mathbf{P}^T\mathbf{y}$, (4) can be written as,

$$\mathbf{Ba} = \mathbf{c}, \tag{5}$$

where $\mathbf{B}$ is an $N * N$ symmetric matrix and $\mathbf{c}$ is an N-dimensional vector. There are well-developed numerical approaches to solve (5). For the case that $\mathbf{P}$ has full rank; i.e., rank N, $\mathbf{P}^T\mathbf{P}$ is nonsingular and positive definite. The normal equations will have a unique solution $\mathbf{a} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{y}$. In this case, the *Cholesky Decomposition*, a fast algorithm which requires only half the usual number of multiplications than alternative methods, can be used to solve (5).

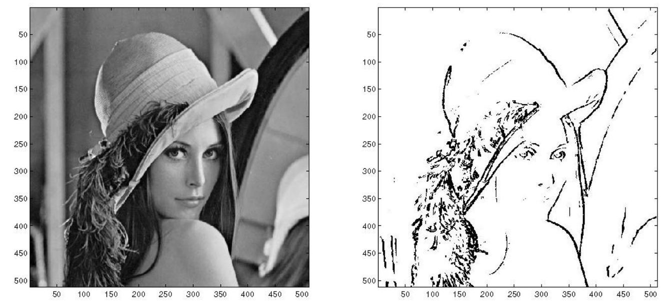
If $\mathbf{P}$ is defective; i.e., rank $< N$, $\mathbf{P}^T\mathbf{P}$ fails to be positive definite and the *Singular Value Decomposition* (SVD) provides the key to solve (5). Indeed, the positive definite property of $\mathbf{B}$ can be easily examined in the process of *Cholesky Decomposition*.

IV. Experiments

In this section, we evaluate the performance of the proposed predictor with edge-look-ahead. Comparisons to existing state-of-the-art predictors and coders are also given. All the test images used in the experiments are from the website of TMW[1] [6]. For LS adaptation, we use the same parameters as defined in EDP [2]; that is, the same training area and the same error threshold. We present the bit rate performance of the system. Finally, we give a description about the computational complexity of the proposed system.

Performance of the proposed system

First of all, we use the image "Lennagrey" (Fig. 2.(a)) to demonstrate the effectiveness of the proposed edge detector. As can be seen in Fig. 2.(b), the pixels around edges in image "Lennagrey" have been picked out successfully.



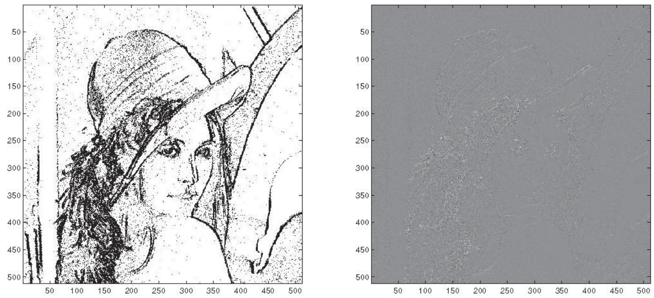
(a) (b)

Fig. 2. (a) The image "Lennagrey". (b) Pixels for which (1) is satisfied in the image "Lennagrey".

The usefulness of the proposed predictor with edge-look-ahead can be demonstrated through the following experiment. We construct two tenth-order LS based predictors; one with the use of the proposed edge-look-ahead mechanism and the other performs LS adaptation in a pixel-by-pixel manner. Then we compare the performance of the two predictors. Again, the image "Lennagrey" in Fig. 2.(a) is used for this experiment.

For the predictor with edge-look-ahead, the pixels for which LS adaptation is activated are shown in Fig. 3.(a). Overall, about 17% of pixels activate the LS adaptation process. The image of uncompensated prediction errors and the corresponding histogram are shown in Fig. 3.(b) and Fig. 4 respectively. As can be seen in Fig. 3.(b), the proposed mechanism performs very well around edges. For comparison, we also show in

[1]http://www.csse.monash.edu.au/~bmeyer/tmw/



(a) (b)

Fig. 3. (a) Pixels for which LS adaptation is used in the image "Lennagrey". (b) Image of uncompensated prediction errors for "Lennagrey".

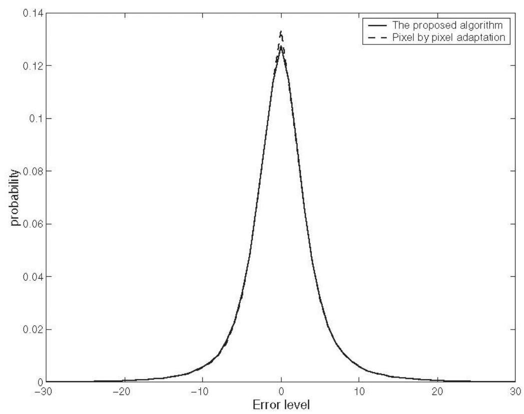


Fig. 4. Histogram of prediction errors for the proposed approach and that of a pixel-by-pixel adaptation.

Fig. 4 the histogram of uncompensated prediction error when the LS adaptation process is performed in a pixel-by-pixel manner. The histogram using the proposed approach is very close to that with pixel-by-pixel adaptation although only 17% of pixels activate the LS adaptation process. The proposed approach has made a good tradeoff between the prediction results and the computational complexity. Indeed, the entropies corresponding to the two histograms in Fig. 4 are respectively 4.159 bits (Proposed approach) and 4.145 bits (adapted in a pixel-by-pixel manner).

Comparisons to existing state-of-the-art predictors

Table I gives comparisons of uncompensated prediction errors for a set of eight test images in first order entropies. To have a comparison with the existing linear and nonlinear predictors, we have completed a set of predictors with different orders from 4 to 10. The results of a median edge detector (MED) [5], a gradient adjusted predictor (GAP) [4] and an edge directed predictor (EDP) with different orders are taken from [2]. As can be seen in Table I, the proposed system can remove the statistical redundancy efficiently. It achieves noticeable improvement when compared with MED and GAP predictor. The proposed predictor also gives lower entropies when compared with those of EDP [2]. Moreover, the results of the proposed approach are very close to those with pixel-by-pixel LS adaptation. This is also why the first two entries of the proposed approach and that with pixel-by-pixel LS adaptation in Table I appear to be identical after taking the ceiling operation to the second decimal point digits.

TABLE I
FIRST ORDER ENTROPIES OF PREDICTION ERRORS.

Image	MED	GAP	EDP				Proposed Algorithm				Pixel by Pixel Optimization			
			N=4	N=6	N=8	N=10	N=4	N=6	N=8	N=10	N=4	N=6	N=8	N=10
Baboon	6.28	6.22	6.04	6.01	6.00	5.99	6.03	5.99	5.98	5.98	6.03	5.99	5.98	5.98
Lena	4.90	4.75	4.64	4.60	4.59	4.58	4.58	4.53	4.53	4.51	4.58	4.53	4.53	4.51
Lennagrey	4.56	4.40	4.32	4.26	4.24	4.22	4.24	4.20	4.19	4.16	4.22	4.18	4.17	4.15
Peppers	4.95	4.78	4.55	4.52	4.51	4.50	4.48	4.45	4.44	4.43	4.47	4.43	4.43	4.43
Barb	5.21	5.15	4.67	4.44	4.40	4.35	4.52	4.36	4.30	4.25	4.46	4.31	4.26	4.21
Barb2	5.19	5.06	4.93	4.80	4.79	4.78	4.90	4.77	4.75	4.75	4.88	4.75	4.74	4.74
Boats	4.31	4.29	4.20	4.14	4.12	4.10	4.16	4.10	4.07	4.05	4.07	4.00	3.97	3.96
Gold Hill	4.72	4.70	4.64	4.60	4.59	4.58	4.64	4.60	4.59	4.59	4.63	4.58	4.57	4.57
Average	5.02	4.92	4.75	4.67	4.66	4.64	4.69	4.63	4.61	4.59	4.67	4.60	4.58	4.57

TABLE II
COMPARISONS WITH EXISTING LOSSLESS IMAGE CODERS (IN BITS/SAMPLE, THE PROPOSED APPROACH IS WITH A SIX-ORDER PREDICTOR).

Image	Proposed	JPEG-LS [5]	CALIC [4]	EDP [2]	TMW [6]
Baboon	5.81	6.04	5.88	5.81	5.73
Lena	4.34	4.61	4.48	4.40	4.30
Lennagrey	3.94	4.24	4.11	4.02	3.91
Peppers	4.26	4.51	4.42	4.35	4.25
Barb	4.11	4.69	4.32	4.11	4.09
Barb2	4.52	4.69	4.53	4.52	4.38
Boats	3.72	3.93	3.83	3.80	3.61
Gold Hill	4.36	4.48	4.39	4.39	4.27
Average	4.38	4.65	4.50	4.43	4.32

To compare with state-of-the-art lossless coders, we also complete a sixth-order coder. We borrow the bias cancellation techniques in [7] so that the prediction is further refined through context modeling. The refined error signal is then entropy encoded using conditional arithmetic coding [8]. Table II gives the actual bit rates by JPEG-LS [5], CALIC [4], EDP [2] and TMW [6] for a set of eight test images. Results listed in the last three columns of Table II are taken directly from [2] and those of the JPEG-LS are simulated with the program from the website of LOCO-I [5]. All the bit rates reported by the proposed algorithm are obtained using the same parameters described in previous sections and no individual optimization is performed. Table II shows that the proposed system achieves lower bit rates than JPEG-LS [5], CALIC [4], EDP [2] and provides competitive results with the highly complex two-pass coder TMW [6].

TABLE III

PERCENTAGE OF PIXELS PERFORMING LS ADAPTATION.

Image	N=4	N=6	N=8	N=10
Baboon	65.1	64.7	64.7	64.5
Lena	24.4	23.9	23.9	23.4
Lennagrey	18.2	17.9	17.9	17.4
Peppers	19.0	18.4	18.3	18.1
Barb	35.8	34.8	34.5	34.3
Barb2	39.7	38.7	38.8	38.8
Boats	19.3	18.8	18.5	18.3
Gold Hill	24.8	24.2	24.0	23.9
Average	30.8	30.2	30.1	29.9

Computational complexity

We show in Table III the percentage of pixels performing LS adaptation. When compared with the predictor performing LS adaptation in a pixel-by-pixel manner, the proposed approach has made a noticeable reduction in computational complexity (Table III) with only a minor degradation in the prediction result (Table I).

Numerically, the normal equations ((4),(5)) can be solved by *Cholesky Decomposition* or *SVD* depending on the rank of $\mathbf{P}$ in (3). For $\mathbf{P}$ to be full-ranked, the *Cholesky Decomposition* can be used and it requires only $N^3/6$ multiplications to solve (5), which is about half the usual number of multiplications than alternative methods. If $\mathbf{P}$ is defective, *SVD*, which requires much higher computations, is applied. Fortunately, our experiments show that most of the LS adaptations in the coding process are solved by the use of *Cholesky Decomposition*. This is because pixels around boundaries usually have large variation in the gray level and thus the matrix $\mathbf{P}$ in (3) is seldom defective. Therefore, most of the computations take place in forming the normal equations (5) rather than solving them. For this, [3] had proposed an inclusion and exclusion method for fast construction of the $\mathbf{P}^T\mathbf{P}$ matrix.

TABLE IV

OPERATION COUNTS FOR EDGE DETECTOR IN (1).

Operation	Compare	ADD/SUB	MUL/DIV	Square
Edge detection	n+2	≤ 4n	≤ 7	≤ (n+3)
"n" is the number of pixels in texture context. In this paper, n=4				

The operation counts for each coding pixel in the edge detection process are listed in Table IV. It should be noted that there is no need to check both of the two inequalities in (1) for every pixel. Only when the variance inequality holds then we check the other condition. Therefore, the actual computational cost is lower than what is listed in Table IV. The execution time (in seconds) of the proposed algorithm and that of pixel-by-pixel adaptation for different orders of predictors are listed in Table V. The proposed approach has achieved a noticeable improvement on the runtime performance with only a minor degradation in entropy (Table I).

TABLE V

COMPARISONS OF THE EXECUTION TIME BETWEEN THE PROPOSED ALGORITHM AND THOSE OF PIXEL-BY-PIXEL ADAPTATION (IN SECONDS, ON A P-III 600MHZ MACHINE).

Image	Proposed Algorithm				Pixel-by-Pixel Adaptation			
	N=4	N=6	N=8	N=10	N=4	N=6	N=8	N=10
Baboon	2.30	7.54	15.41	22.49	3.41	11.53	23.20	33.98
Lena	0.99	3.06	6.04	8.55	3.43	11.49	23.54	57.58
Lennagrey	0.83	2.38	4.58	6.41	3.49	11.18	23.36	34.59
Peppers	0.84	2.40	4.68	6.63	3.30	11.24	23.02	33.66
Barb	1.46	4.18	8.48	12.25	3.35	11.28	23.08	34.02
Barb2	2.45	7.56	14.91	21.93	5.72	18.00	59.46	90.95
Boats	1.33	3.84	7.34	10.53	5.38	17.73	36.63	53.80
Gold Hill	1.61	4.85	9.61	13.61	5.70	17.70	59.17	90.97
Average	1.48	4.48	8.88	12.80	4.22	13.76	33.93	53.69

V. CONCLUSION

In this paper, an LS-based adaptive predictor for lossless image coding has been proposed. By exploiting the edge-directed characteristic of LS-based predictor, we propose initiating the LS adaptation process only when the coding pixel is around an edge or when the prediction error is greater than a predefined threshold. For this, a simple yet effective edge detector using only causal pixels is proposed. With the proposed edge detector, the predictor can look ahead if the coding pixel is around an edge and initiate the LS adaptation process beforehand to prevent the occurrence of a large prediction error. When compared with the pixel-by-pixel LS adaptation, the proposed approach can achieve a noticeable reduction in complexity with only a minor degradation in entropy; a good tradeoff between computational complexity and the prediction results has been obtained.

REFERENCES

[1] I. Avcibas, N. Memon, B. Sankur, and K. Sayood, ‘A Successively Refinable Lossless Image-Coding Algorithm,” *IEEE Trans. Commun.*, Vol. 53, No. 3, pp. 445-452, Mar. 2005.

[2] Xin Li, and Michael T. Orchard, “Edge-directed prediction for lossless compression of natural images,” *IEEE Trans. Image Process.*, Vol. 10, No. 6, pp. 813-817, June 2001.

[3] N. Kuroki, T. Nomura, M. Tomita and K. Hirano, “Lossless image compression by two-dimensional linear prediction with variable coefficients,” *IEICE Trans. Fund.*, Vol. E75-A, No. 7, pp. 882-889, July 1992.

[4] X. Wu, and N. Memon, “Context-based, adaptive, lossless image coding,” *IEEE Trans. Commun.*, Vol. 45, No. 4, pp. 437-444, Apr. 1997.

[5] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Trans. Image Process.*, Vol. 9, No. 8, pp. 1309-1324, Aug. 2000. http://www.hpl.hp.com/loco/locodownold.htm

[6] B. Meyer, and P. E. Tischer, “TMW-A new method for lossless image compression,” in *Proc. Int. Picture Coding Symp.*, Berlin, Germany, Oct. 1997. http://www.csse.monash.edu.au/∼bmeyer/tmw/paper.ps

[7] X. Wu, “Lossless compression of continuous-tone images via context selection, quantization, and modeling,” *IEEE Trans. Image Process.*, Vol. 6, No. 5, pp. 656-664, May 1997.

[8] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Commun. of the ACM*, Vol. 30, No. 6, pp. 520-540, June 1987.