Adaptive Lossless Image Coding Using Least Squares Optimization With Edge-Look-Ahead

Lih-Jen Kau and Yuan-Pei Lin, *Senior Member, IEEE*

***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.**

***Index Terms*—Adaptive prediction, context modeling, edge detection, entropy coding, least squares (LS) optimization, lossless image coding.**

I. Introduction

MANY OF THE recent advances in lossless image coding are based on predictive coding with context modeling [1]–[8]. 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 [1]–[4]. Among which, edge-directed prediction (EDP) [1] 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 [1] 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 [1] has made a noticeable improvement over the state-of-the-art lossless image coder—context-based adaptive lossless image coding (CALIC) [5].

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 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. 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.

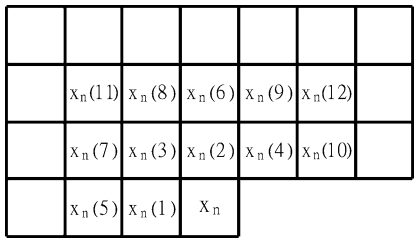

Fig. 1. Ordering of pixels for prediction inputs.

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 noncausal 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 $x_n(1), \ldots, x_n(4)$ in Fig. 1

$$\kappa = \{x_n(1), \ldots, x_n(4)\}. \tag{1}$$

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.

Manuscript received December 3, 2004; revised February 23, 2005. This work was supported by the NSC under Grant 93-2213-E-009-115. This paper was recommended by Associate Editor A. Loui.

The authors are with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan, R.O.C. (e-mail: u8912801@cc.nctu.edu.tw; ypl@cc.nctu.edu.tw).

Digital Object Identifier 10.1109/TCSII.2005.852194

A pixel around an edge is likely to have a large σ^2 but small σ_h^2 and σ_l^2. We determine whether the coding pixel is around an edge if the following two conditions are both satisfied:

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

where 0.01 is added so that the denominator of (2) does not become 0 when σ_h^2 and σ_l^2 are both zero. The case that σ_h^2 and σ_l^2 are both zero can occur in an artificial image. 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. 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

$$\hat{x}_n = \sum_{k=1}^{N} a(k) x_n(k) \tag{3}$$

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 (2) 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

$$\mathbf{Pa} = \mathbf{y} \tag{4}$$

where

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

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

To minimize the square errors, $\|\mathbf{y} - \mathbf{Pa}\|_2^2$, for (4), the vector $\mathbf{a}$ should satisfy the normal equations [10]

$$\mathbf{P}^T\mathbf{Pa} = \mathbf{P}^T\mathbf{y}. \tag{5}$$

If we define $\mathbf{B} = \mathbf{P}^T\mathbf{P}$ and $\mathbf{c} = \mathbf{P}^T\mathbf{y}$, (5) can be written as

$$\mathbf{Ba} = \mathbf{c} \tag{6}$$

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 (6). 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 (6) [10], [11].

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 (6) [10]. Indeed, the positive definite property of $\mathbf{B}$ can be easily examined in the process of *Cholesky decomposition* [11].

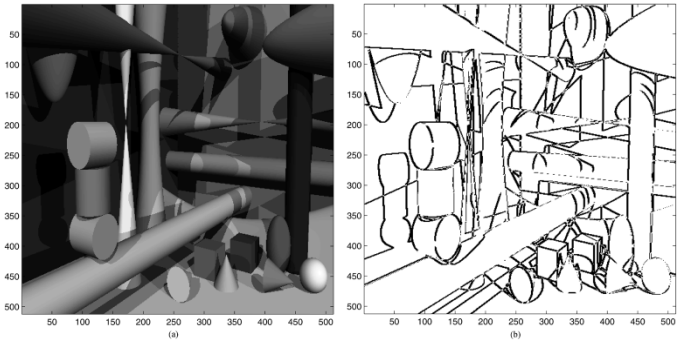

Fig. 2. (a) Image "Shapes." (b) Pixels for which (2) is satisfied in the image "Shapes."

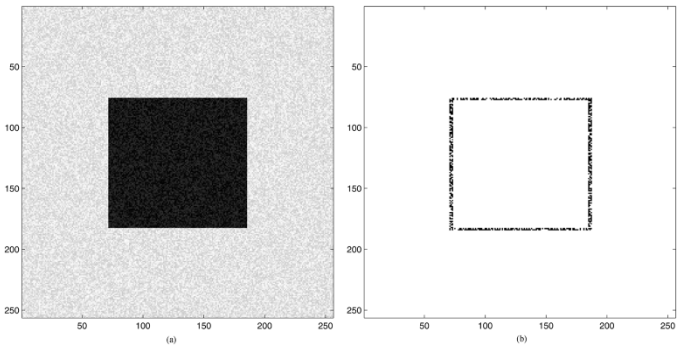

Fig. 3. (a) Image "Noisesquare." (b) Pixels for which (2) is satisfied in the image "Noisesquare."

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][7]. For LS adaption, we use the same parameters as defined in EDP [1] ; that is, the same training area and the same error threshold. We first demonstrate the usefulness of the proposed edge detector and then present the bit rate performance of the system. Finally, we give a description about the computational complexity of the proposed system.

The Edge Detector

To demonstrate the effectiveness of the proposed edge detector, we use the image "Shapes" [Fig. 2(a)], which is an artificial image with many edges and lines. The pixels that satisfy

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

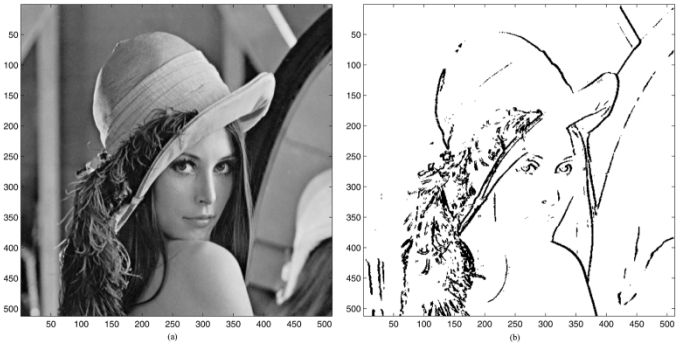

Fig. 4. (a) Image "Lennagrey." (b) Pixels for which (2) is satisfied in the image "Lennagrey."

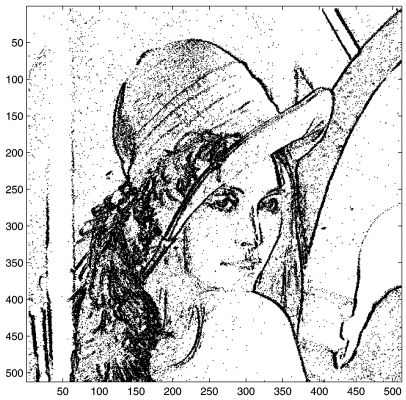

Fig. 5. Pixels for which LS adaption is used in the proposed edge-look-ahead predictor for the image "Lennagrey."

the two conditions in (2) are marked in Fig. 2(b). We can see from Fig. 2(b) that the edge detector has successfully picked out the pixels around edges. To test the robustness of the detector, we apply the edge detector to the image "Noisesquare" [Fig. 3(a)], an image with salt-and-pepper noise. The pixels picked out by the edge detector are as shown in Fig. 3(b). We see from Fig. 3(b) that the edge detector is robust to moderate salt-and-pepper noise. In addition to artificial images, we also apply the edge detector to natural image, "Lennagrey" [Fig. 4(a)]. As can be seen in Fig. 4(b), the pixels around edges in image "Lennagrey" have been picked out successfully.

Performance of the Proposed System

The usefulness of the proposed predictor with edge-look-ahead mechanism 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. The image "Lennagrey" in Fig. 4(a) is used for this experiment.

For the predictor with edge-look-ahead, the pixels for which LS adaption is activated are shown in Fig. 5. Overall, about 17% of pixels activate the LS adaptation process. The image of uncompensated prediction errors and the corresponding histogram are shown in Figs. 6 and 7 respectively. As can be seen in Fig. 6, the proposed mechanism performs very well around edges. For comparison, we also show in Fig. 7 the

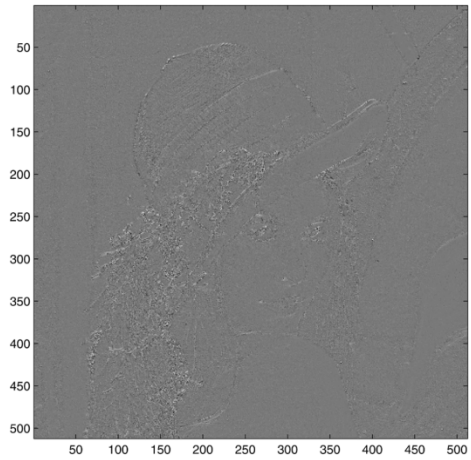

Fig. 6. Image of uncompensated prediction errors using the proposed edge-look-ahead approach for "Lennagrey."

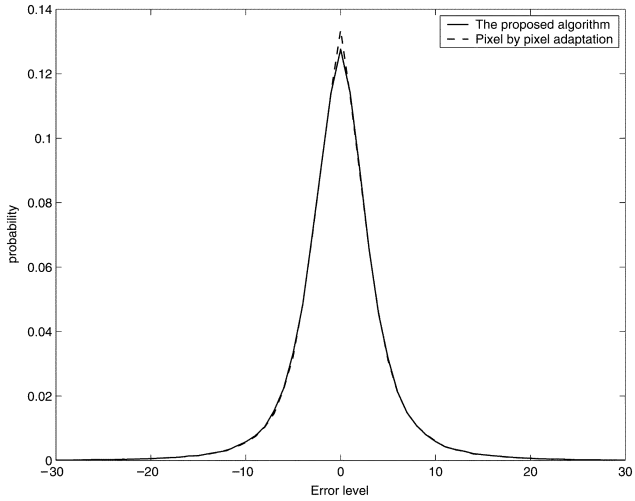

Fig. 7. Histogram of prediction errors for the proposed approach and that of a pixel-by-pixel adaptation.

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 achieved a good tradeoff between the prediction results and the computational complexity. Indeed, the entropies corresponding to the two histograms in Fig. 7 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) [6], a gradient adjusted predictor (GAP) [5] and an EDP with different orders are taken from [1]. As can be seen in Table I, the proposed system can remove the statistical redundancy efficiently. It achieves noticeable improvement when compared with MED

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 SECOND COLUMN IS EXECUTION TIME OF PROPOSED EDGE-LOOK-AHEAD APPROACH ON A PIII-600 MHZ MACHINE

Image	Proposed	seconds	JPEG-LS [6]	CALIC [5]	EDP [1]	TMW [7]
Baboon	5.81	8.44	6.04	5.88	5.81	5.73
Lena	4.34	6.02	4.61	4.48	4.40	4.30
Lennagrey	3.94	5.87	4.24	4.11	4.02	3.91
Peppers	4.26	5.93	4.51	4.42	4.35	4.25
Barb	4.11	7.41	4.69	4.32	4.11	4.09
Barb2	4.52	11.19	4.69	4.53	4.52	4.38
Boats	3.72	9.48	3.93	3.83	3.80	3.61
Gold Hill	4.36	9.32	4.48	4.39	4.39	4.27
Average	4.38	7.96	4.65	4.50	4.43	4.32

TABLE III
PERCENTAGE OF PIXELS PERFORMING LS ADAPTION AND NUMBER OF PIXELS PERFORMING *CHOLESKY DECOMPOSITION* AND SVD

Image	Proposed linear predictor with edge-look-ahead											
	N=4			N=6			N=8			N=10		
	%	Cholesky	SVD	%	Cholesky	SVD	%	Cholesky	SVD	%	Cholesky	SVD
Baboon	65.1	170545	0	64.7	169588	0	64.7	169651	0	64.5	169203	0
Lena	24.4	63989	16	23.9	62681	52	23.9	62568	76	23.4	61136	83
Lennagrey	18.2	47757	14	17.9	46856	29	17.9	46754	42	17.4	45607	32
Peppers	19.0	49789	0	18.4	48362	0	18.3	47940	0	18.1	47567	0
Barb	35.8	93825	0	34.8	91254	0	34.5	90552	0	34.3	89946	0
Barb2	39.7	164570	0	38.7	160509	0	38.8	160888	0	38.8	160829	0
Boats	19.3	79992	65	18.8	78030	0	18.5	76863	21	18.3	76036	36
Gold Hill	24.8	102857	1	24.2	100169	0	24.0	99628	0	23.9	99086	0
Average	30.8	96666	12.0	30.2	94681	10.1	30.1	94356	17.4	29.9	93676	18.9

and GAP predictor. The proposed predictor also gives lower entropies when compared with those of EDP [1]. Moreover, the results of the proposed approach are very close to those with pixel-by-pixel LS adaptation.

To compare with state-of-the-art lossless coders, we also complete a sixth-order coder. We borrow the bias cancellation techniques in [8] so that the prediction is further refined through context modeling. The refined error signal is then entropy encoded using conditional arithmetic coding [9]. Table II gives the actual bit rates by JPEG-LS [6], CALIC [5], EDP [1], and TMW [7] for a set of eight test images. In Table II, the results of CALIC, EDP and TMW are taken directly from [1] and those of the JPEG-LS are simulated with the program from the website of LOCO-I [6]. 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. We also show in the second column of Table II the execution time of the proposed coder so that we can get a picture on the runtime performance of the proposed coder. Table II shows that the proposed system achieves lower bit rates than JPEG-LS [6], CALIC [5], EDP [1] and provides competitive results with the highly complex two-pass coder TMW [7].

Computational Complexity

Numerically, the normal equations [(5) and (6)] can be solved by *Cholesky decomposition* or SVD depending on the rank of $\mathbf{P}$ in (4). For $\mathbf{P}$ to be full-ranked, the *Cholesky decomposition* can be used and it requires only $N^3/6$ multiplications to solve (6), which is about half the usual number of multiplications than alternative methods [10], [11]. 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 can be seen in Table III, where we have listed the percentage

TABLE IV
OPERATION COUNTS FOR EDGE DETECTOR IN (2)

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

TABLE V
COMPARISONS OF EXECUTION TIME BETWEEN PROPOSED ALGORITHM AND THOSE OF PIXEL-BY-PIXEL ADAPTATION (IN SECONDS, ON A PIII-600 MHZ 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

of pixels performing LS adaption and the number of pixels performing *Cholesky decomposition* or *SVD* for predictors with different orders. Indeed, this is because pixels around boundaries usually have large variation in the gray level and thus the matrix $\mathbf{P}$ in (4) is seldom defective. Therefore, most of the computations take place in forming the normal equations (6) rather than solving them. For this, [2] had proposed an inclusion and exclusion method for fast construction of the $\mathbf{P}^T\mathbf{P}$ matrix.

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 (2) 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 when compared with that of pixel-by-pixel adaptation approach (Table I).

V. CONCLUSION

In this paper, an LS-based adaptive predictor for lossless image coding has been proposed. By exploiting the edge-directed characteristic of the 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] L. Xin and M. T. Orchard, "Edge-directed prediction for lossless compression of natural images," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 813–817, Jun. 2001.
[2] 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, Jul. 1992.
[3] H. Ye, G. Deng, and J. C. Devlin, "A weighted least squares method for adaptive prediction in lossless image compression," in *Proc. Picture Coding Symp.*, Saint-Malo, France, 2003, pp. 489–493.
[4] B. Meyer and P. E. Tischer, "Glicbawls—Grey level image compression by adaptive weighted least squares," in *Proc. Data Compression Conf. 2001*, Snowbird, UT, 2001, pp. 503–503.
[5] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.
[6] 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.
[7] B. Meyer and P. E. Tischer. TMW-A new method for lossless image compression. presented at *Proc. Int. Picture Coding Symp.*. [Online]. Available: http://www.csse.monash.edu.au/~bmeyer/tmw/paper.ps
[8] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Trans. Image Processing*, vol. 6, pp. 656–664, May 1997.
[9] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
[10] S. J. Leon, *Linear Algebra With Applications*, NJ: Prentice Hall, 2002, pp. 482–488.
[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge University Press, 2002, pp. 96–98.