Least-Squares-Based Switching Structure for Lossless Image Coding

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

Abstract—Many coding methods are more efficient with some images than others. In particular, run-length coding is very useful for coding areas of little changes. Adaptive predictive coding achieves high coding efficiency for fast changing areas like edges. In this paper, we propose a switching coding scheme that will combine the advantages of both run-length and adaptive linear predictive coding. For pixels in slowly varying areas, run-length coding is used; otherwise least-squares (LS)-adaptive predictive coding is used. Instead of performing LS adaptation in a pixel-by-pixel manner, we adapt the predictor coefficients only when an edge is detected so that the computational complexity can be significantly reduced. For this, we use a simple yet effective edge detector using only causal pixels. This way, the proposed system can look ahead to determine if the coding pixel is around an edge and initiate the LS adaptation in advance to prevent the occurrence of a large prediction error. With the proposed switching structure, very good prediction results can be obtained in both slowly varying areas and pixels around boundaries. Furthermore, only causal pixels are used for estimating the coding pixels in the proposed encoder; no additional side information needs to be transmitted. Extensive experiments as well as comparisons to existing state-of-the-art predictors and coders will be given to demonstrate its usefulness.

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

I. INTRODUCTION

THERE have been great advances in lossless image coding recently [1]–[23], [26]–[31]. Some of which are based on reversible wavelet transformation using lifting structure [6]–[10]. By using integer wavelet transformation, lossless to near-lossless compression as well as progressive reconstruction of image data can be achieved [6]–[10]. However, the compression results obtained with the use of integer wavelet transformation are typically inferior to that of obtained by predictively encoded techniques [11]. Therefore, an approach that achieves progressive transmission of lossless and near-lossless coding of image data using predictive coding in a single frame-

L.-J. Kau is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C., and also with the Department of Computer and Communication Engineering, Dahan Institute of Technology, Hualien 971, Taiwan, R.O.C. (e-mail: u8912801@cc.nctu.edu.tw).

Y.-P. Lin is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: ypl@mail.nctu.edu.tw).

work has been proposed [11]. The results presented in [11] are competitive to that of obtained by state-of-the-art compression schemes.

To accommodate the varying statistics of coding images, adaptive predictors with context modeling are often used in predictive coding schemes [11]–[23], [26]–[31]. Besides, adaptive prediction is achieved in most of the coders by using multi-predictor structures [13]–[23]. Among which, the context-based adaptive lossless image coding (CALIC) system [16], a state-of-the-art lossless coder proposed for JPEG-LS, uses a gradient adjusted predictor (GAP). Based on the gradient of neighboring pixels, one out of a set of seven predictors is chosen. The low-complexity lossless compression for images (LOCO-I) coder [17], an algorithm motivated by CALIC [16] and standardized into JPEG-LS, uses a median edge detector (MED) to choose one of three predictors for current prediction. In [18], adaptive prediction is achieved by choosing one out of a set of predictors that minimizes the energy of prediction errors in a specified cluster of causal pixels and the predictor coefficients of the selected predictor are then updated by applying gradient descent rule. In [19], [20], multiple-pass prediction is introduced. With multiple passes, the encoder can form a 360° prediction [19] or perform a global image analysis [20]. A highly complex two-pass coder called TMW has been proposed in [20]. Using multiple linear predictors and global image analysis, the TMW system can achieve lower bit rates than existing coders for most images. While achieving very low bit rates, the computational cost is regarded as prohibitive in TMW [20]. Recently, a fuzzy logic-based adaptive DPCM algorithm called fuzzy logic-based matching pursuits (FMP) [21] is proposed. The FMP presents a competitive, and in some cases superior result than TMW but with a lower computational cost. Though FMP is effective in removing the statistical redundancy, it still takes minutes.

In the context of optimal predictors, the minimum mean square error estimate of Y given observations $X_1, X_2, \cdots, X_n$ is $E\{Y|X_1, X_2, \cdots, X_n\}$, generally a nonlinear function. Therefore, there have been many results using neural networks as nonlinear estimators [22], [23]. Neural-network-based predictors perform well in slowly varying areas. However, there can be large prediction error around boundaries [24]. The result can be improved using additional hidden layers or hidden neurons, but this incurs a drastic increase in complexity [22], [25].

Recently, linear predictors adapted by LS optimization have been proposed as an efficient approach to accommodate varying statistics of coding images [26]–[31]. Among which, the edge-directed prediction (EDP) [26] pointed out that the superiority
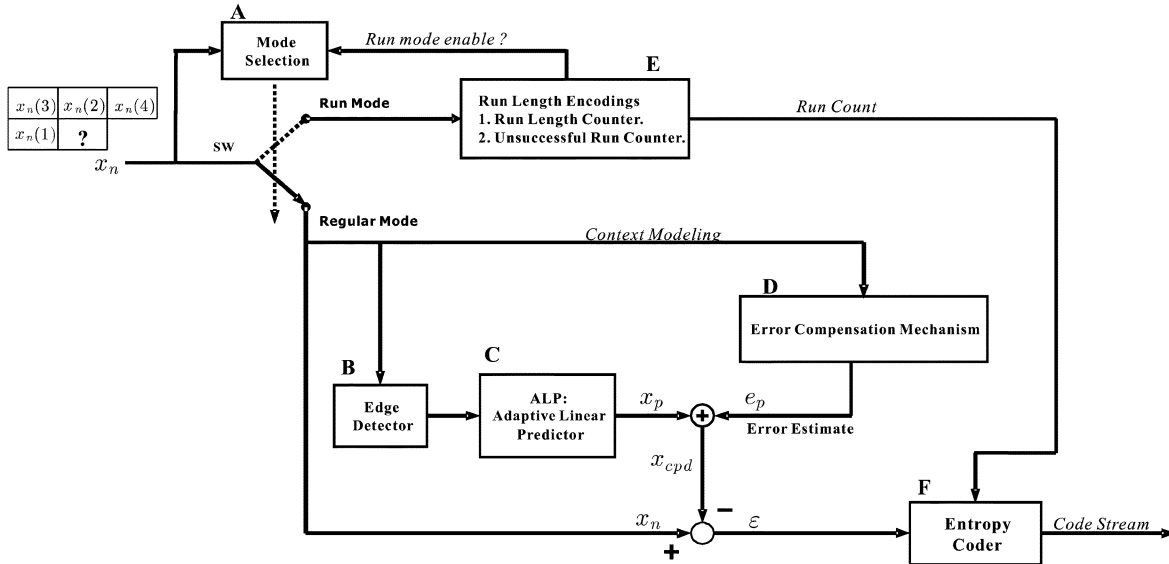
Fig. 1. Proposed RALP system.

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 [26] 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 [26] has made a noticeable improvement over the state-of-the-art lossless coder CALIC.

It is known that many coding methods are more efficient with some images than others. In particular, run-length coding is very useful for coding areas of little changes. Adaptive predictive coding achieves high coding efficiency for fast changing areas like edges. In this paper, we propose a switching coding scheme (as shown in Fig. 1) that will combine the advantages of both run-length and adaptive linear predictive (RALP) coding. There are other switching methods that achieve very low bit rates [20]–[22]. However the results are usually obtained with a very high computational complexity [20]–[22]. On the contrary, the proposed RALP coder can achieve a very good coding efficiency but still with a moderate computational complexity. In the proposed approach, the run-length encoder is used for pixels in slowly varying areas; otherwise an LS-based adaptive predictor is used. The LS-based predictor has been shown to be very useful for the prediction of pixels around an edge [26], [27]. Moreover, we adapt the predictor coefficients only when an edge is detected or when the prediction error is beyond a pre-selected threshold so that the computational cost can be significantly reduced [27]. To do this, we use a simple and efficient edge detector that uses only causal pixels, i.e., pixels that have already been coded. This way, 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. With the proposed switching structure, very good prediction results can be obtained in both slowly varying areas and pixels around boundaries. Some preliminary results regarding the proposed LS-based predictor with edge-look-ahead can be found in [27].

For prediction error refinement, the so-called context modeling technique [16], [19] is used in the proposed system. The compensated error has a narrower histogram and hence a lower first-order entropy. As we will see in the experiments that the switching structure combined with edge-look-ahead prediction as well as automatic error modeling renders the proposed RALP highly adaptable and very feasible under limited resources. A very good trade-off between coding efficiency and computational complexity can be achieved. Comparisons with existing state-of-the-art LS-based predictors can also be found in our experiments.

The rest of the paper is organized as follows. Section II gives an overview of the proposed RALP system. Section III introduces the proposed LS-adaptive predictor. The error compensation is given in Section IV. Section V addresses the entropy coding of prediction error. Extensive experiments of the proposed method and comparisons to existing predictors and coders are given in Section VI. A conclusion is given in Section VII.

II. PROPOSED RALP SYSTEM

The proposed RALP system, as shown in Fig. 1, has two operation modes, run mode and regular mode. The "*mode selection*" block (Fig. 1) determine if the current pixel is in an local area of little changes. If it is, the run mode is triggered and the current pixel is encoded using run-length encoding. If not, the regular mode is assumed and the pixel is encoded using predictive coding.

A. Run Mode

It is known that the run-length coding is most efficient for the encoding of consecutive pixels with identical gray values. The case that consecutive pixels are identical can usually occur in an artificial image or in slowly varying areas of a natural image. Therefore, we use the run-length coding in the proposed RALP system for the encoding of pixels in an area of little changes. If the four pixels $x_n(1), \cdots, x_n(4)$ in Fig. 2 are identical, the run mode is switched on and the run-length is encoded using an

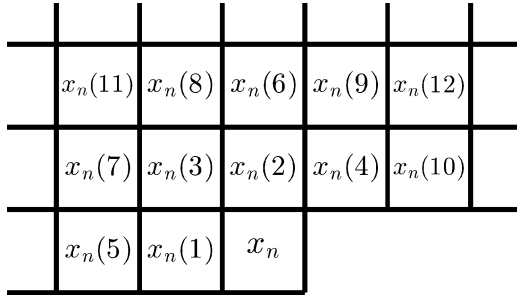| $x_n(11)$ | $x_n(8)$ | $x_n(6)$ | $x_n(9)$ | $x_n(12)$ |
| $x_n(7)$ | $x_n(3)$ | $x_n(2)$ | $x_n(4)$ | $x_n(10)$ |
| $x_n(5)$ | $x_n(1)$ | x_n | | |

Fig. 2. Ordering of pixels for prediction inputs.

arithmetic coder with an alphabet set of $\{0, 1, \cdots, 20\}$. The 0, called escape symbol in this paper, is used to indicate an unsuccessful run and should be encoded if the gray value of the coding pixel x_n and that of $x_n(1), \cdots, x_n(4)$ are distinct so that the decoder can also make a right decision and quit the run mode automatically. This time, the regular mode is used for the encoding of the current pixel. It is noted that the run mode can also be broken by ends of lines, in which case the encoder returns to the regular mode, i.e., the regular mode is assumed for the first pixel of every line. Moreover, the encoding of an escape symbol can cause penalty and degrade the coding efficiency. Therefore, we record the number of times of run mode triggered and the times of unsuccessful run. If the percentage of unsuccessful run is greater than a predefined threshold, the run mode is disabled and not to be used for the rest of the coding process. It is noted that all the pixels used for mode selection are causal and the decoder can reproduce the same decisions without any side information.

B. Regular Mode

In the regular mode, pixels are encoded using predictive coding. Predictive coding can be very efficient for the removal of statistical redundancy between neighboring pixels in slowly varying areas. However there can have a large prediction error around boundaries. In this paper, we will use LS optimization to update the predictor coefficients on the fly so that the predictor can adapt itself to the varying statistics [27]. It is known that the LS-based adaptive predictor is an efficient approach to improve the prediction result around boundaries for its edge-directed property [26], [27]. However a pixel-by-pixel adaptation of the predictor coefficients is computationally expensive and often not necessary. Therefore, we will initiate adaptation only when the prediction is inadequate, which is around an edge. For this, an "*edge detector*" is used to look ahead and determine if the coding pixel is around an edge so that the predictor can adapt itself beforehand to prevent the occurrence of a large prediction error. In the regular mode, the prediction error is further refined using *error compensation*. That is, the predictor output x_p is added by a correction term e_p (as shown in Fig. 1) to get a compensated prediction $x_{cpd} = x_p + e_p$. The amount of compensation e_p is determined through an error modeling mechanism. The refined error signal $\varepsilon = x_n - x_{cpd}$ can then be entropy encoded using conditional arithmetic coding to produce the coded bit stream.

In the proposed encoder, only causal pixels, i.e., pixels that have already been coded, are used for estimating the coding

pixels; no additional side information needs to be transmitted. Moreover, the proposed RALP coder is symmetric, meaning that the decoder has the same predictor switch as the encoder, and performs prediction and error compensation just like the encoder. Therefore, the actual pixel value can be reconstructed in the decoder with the received bit stream of refined errors. Details of the individual components of the system are introduced in subsequent sections.

III. LS-ADAPTIVE PREDICTION

In the regular mode, we use an LS-based adaptive predictor to accommodate the varying statistics of the image. To save computations, the predictor is adapted only when prediction error is large or likely to be large. For a pixel around an edge, prediction error is usually large and adaptation is needed. To determine whether the coding pixel is around an edge, we use a simple yet effective edge detector in this paper. It should be noted that conventional edge detectors, e.g., "Sobel" operator, can not be applied here because they use non-causal pixels, i.e., pixels yet to be encoded.

We observe that an area that contains an edge usually has a large variance. 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. Moreover, the set of the four pixels $\kappa = \{x_n(1), \cdots, x_n(4)\}$ in Fig. 2 are used for the detection. The mean $\bar{x}$ and variance σ^2 of the set κ are calculated. Furthermore, the four pixels can be divided into two groups, the pixels with gray levels higher than $\bar{x}$ in group κ_h and the rest in group κ_l. We also compute the respective variance σ_h^2, σ_l^2 of the pixels in κ_h and κ_l.

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:

$$\sigma^2 \geq \gamma_1 \quad \text{and} \quad \sigma^2 \geq \gamma_2 \left(\sigma_h^2 + \sigma_l^2 \right). \tag{1}$$

It is noted that the second condition in (1) is included because a region with uniformly distributed gray values also results in a large σ^2. Therefore, the switch first examines if $\sigma^2 \geq \gamma_1$ when a large σ^2 is detected then the switch checks the second inequality in (1). In this paper, the LS adaptation process in the regular mode is activated whenever the two conditions in (1) are satisfied. 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. It should also be noted that the run mode will be triggered when $\sigma^2 = 0$, i.e., the case that $x_n(1), \cdots, x_n(4)$ are identical. In this case, we do not have to check the conditions in (1). As we will see later in experiments that the proposed detector is very effective in detecting edges although only four pixels are used. Moreover, since we use only causal pixels for the detection of an edge, the decoder can perform the same edge detection operation and switches on the LS adaptation process.

In the regular mode, the LS adaptation process is activated whenever the two conditions in (1) are both satisfied or when the prediction error is greater than a predefined threshold θ. The corresponding predictor inputs for different prediction orders
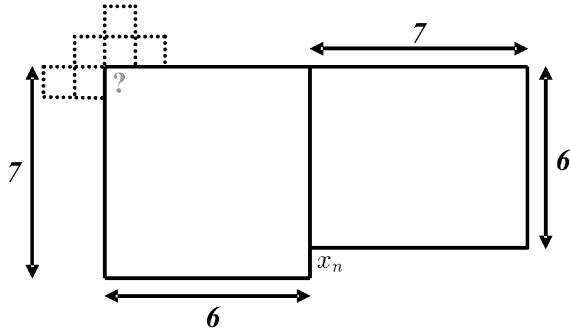
Fig. 3.   Online training regions for the proposed predictor.

are shown in Fig. 2 where the ordering of pixels is based on the distance to the pixel to be encoded. In this paper, the predicted value x_p of the coding pixel x_n is a linear combination of its causal neighbors given by

$$x_p = \sum_{k=1}^{N} a(k)x_n(k) \qquad (2)$$

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. It is noted that we do not use any training set for the optimization of initial predictor coefficients in this paper. The initial coefficients for the proposed predictor are equally weighted, i.e., the coefficients $a(k)$ for the Nth-order predictor are $1/N$ respectively.

The training area for LS adaptation process of the coding pixel is shown in Fig. 3. Suppose we have M pixels in the training area, our objective is to find a LS solution for the system

$$\mathbf{Pa} = \mathbf{y} \qquad (3)$$

where

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

$$\mathbf{a} = \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(N) \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} x_{n-1} \\ x_{n-2} \\ \vdots \\ x_{n-M} \end{bmatrix}.$$

The optimal $\mathbf{a}$ that minimizes the square errors $\|\mathbf{y} - \mathbf{Pa}\|_2^2$ can be obtained by solving the normal equations [33]

$$\mathbf{P}^T\mathbf{Pa} = \mathbf{P}^T\mathbf{y}. \qquad (4)$$

There are well-developed numerical approaches to solve (4). For the case that $\mathbf{P}$ has full rank; i.e., rank N, $\mathbf{P}^T\mathbf{P}$ is non-singular and positive definite [33], [34]. 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*, which requires only half the usual number of multiplications than alternative methods, can be used to solve (4) [33], [34].

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) can be used to solve (4) [33], [34]. The positive definite property of $\mathbf{P}^T\mathbf{P}$ can be easily examined in the process of *Cholesky decomposition* [34].

In this paper, updated predictor coefficients are used for current prediction and passed on to the next coding pixel. For non-edge pixels, we use the stored prediction coefficients of the four nearest causal neighbors to generate four prediction values and take their average as the final prediction result. This manner, the predictor can resist against moderate salt-and-pepper noise. To summarize, the proposed algorithm is given in the following pseudo code.

{**Pseudo code for the proposed approach**.}

runmode = 0

while (scan unfinished)

{

    if (runmode or $(\sigma^2 = 0)$)

    /∗ the case that run mode is initiated ∗/

        {

            if $(x_n = x_{(n-1)})$

                {

                    runmode = 1;

                    runlength + +;

                }

            else

                {

                    Encode runlength using arithmetic coding.

                    runlength = 0;

                    runmode = 0;

                }

        }

    else

        runmode = 0;

    if (not runmode)

    /∗ the case that regular mode is used ∗/

        {

            edge_detected = 0;

            if $(\sigma^2 \geq \gamma_1)$

```
        {
                Calculate variances σ²_h and σ²_l;

                if (σ² ≥ γ₂(σ²_h + σ²_l))

                /* the case that an edge exists */

                        edge_detected = 1;

        }

        if (edge_detected or (e_(n−1) ≥ θ))

                Adapt predictor;

        x_p = Σ_{k=1}^N a(k)x_n(k); /* Perform
        prediction */

        Perform error compensation;

        Encode the refined error signal ε using
        conditional arithmetic coding;

        } /* end of regular mode */

}/* end of scan */
```

IV. ERROR COMPENSATION

It is known that the prediction error $x_n - x_p$ in the regular mode can be further refined by learning from previous predictions, i.e., the so-called bias cancelation technique [16], [19]. To do this, we define the *compound context* v of a coding pixel as

$$v = \{x_n(1), \cdots, x_n(6), e_n(1), \cdots, e_n(4)\} \qquad (5)$$

where $x_n(i)$, $i = 1, \cdots, 6$ are as shown in Fig. 2 and $e_n(i)$, $i = 1, \cdots, 4$ are the uncompensated prediction errors corresponding to $x_n(1), \cdots, x_n(4)$, respectively. We have incorporated prediction errors in error modeling because the amount of compensation is likely to be related to the prediction errors of neighboring pixels.

In the proposed RALP system, error modeling is achieved by performing context clustering with a fixed number of contexts. For this, a set of initial contexts are generated off-line using the image "Lennagrey" (Fig. 4) for the clustering process. For each coding pixel, the compound context v is assigned to one of the existing contexts using mean absolute error distance measure and the corresponding context is then modified accordingly in the coding process. By classifying coding pixels with similar context into the same group, the amount of compensation e_p for the current prediction can be estimated by calculating the sample mean of prediction errors in that group. Therefore, the value e_p to be used in compensating the current prediction is given by

$$e_p = \frac{S}{N} \qquad (6)$$

where S is the prediction errors accumulated in the context which v belongs to, and N is the number of members in that context. With the correction term e_p, we now form a more refined prediction $x_{\text{cpd}} = x_p + e_p$. The compensated error $\varepsilon =$

Fig. 4. Image "Lennagrey."

$x_n - x_{\text{cpd}}$ has a narrower histogram and hence a lower first-order entropy. In the regular mode, the refined error ε is then entropy encoded using a conditional arithmetic coder to produce the bit stream.

V. ENTROPY CODING OF PREDICTION ERROR

For entropy coding of the refined error signal ε, we borrow some of the concepts in [16], [17], [19] so that the coding efficiency in actual bit rates can be further improved.

A. Conditional Entropy Coding

It is known that the coding efficiency can be further improved with the use of conditional probability models. By classifying similar prediction errors in the same group, the coding efficiency can be improved by sharpening the error histogram in each group and thus a smaller conditional entropy can be obtained. To perform the classification, we have to define an error strength estimate Δ so that both the encoder and decoder know exactly which group does the prediction error belongs to or which probability model should be used. In this paper, we define the error strength estimate of the coding pixel to be $|e_p|$, that is

$$\Delta = |e_p|. \qquad (7)$$

It is noted that the e_p, introduced in the previous section, is used in compensating the prediction error. Moreover, it is also available in both the encoder and the decoder. Therefore, we have applied the use of $|e_p|$ as the error strength estimate.

By conditioning on the error strength estimate Δ, we can quantize refined errors into classes of different variances [16], [19], [26]. Moreover, the error histogram in each quantization bins can be sharpened and thus a smaller conditional entropy can be obtained. Therefore, we are using the conditional probability model $P(\varepsilon|\Delta)$ instead of $P(\varepsilon)$ for entropy coding of refined errors. Furthermore, to find the optimal number of quantization bins as well as an optimal quantization for Δ so that the

conditional entropy can be minimized, we define a cost function $C(Q, \varepsilon, \Delta)$ as

$$
\begin{aligned}
C(Q, \varepsilon, \Delta) &= E\left\{H\left(\varepsilon | Q(\triangle)\right)\right\} \\
&= -\sum_{Q(\triangle)=1}^{N} P(Q(\triangle)) \sum_{\varepsilon} P(\varepsilon | Q(\triangle)) \\
&\quad \times \log_2 \left[P(\varepsilon | Q(\triangle))\right]
\end{aligned} \tag{8}
$$

where $Q(.)$, which maps Δ into one of the N quantization bins, is the quantizer to be designed and N is the number of quantization bins to be decided. With such cost function, we are calculating the first-order entropy of ε assigned in each quantization bins, i.e., $H(\varepsilon | Q(\triangle))$, and then take expectation on the entropy $H(\varepsilon | Q(\triangle))$.

To minimize the cost function in (8), we use a set of training pairs (ε, Δ) recorded in the prediction process of fourteen test images [20] in next section. By performing dynamic programming off-line, the final number of quantization bins is found to be three and the optimized quantization of Δ is found to be [0, 1], (1, 55], (55, ∞). Though the quantization may be suboptimal, the compression result is satisfactory as the simulation results in Section VI will demonstrate. Using the above quantization bins, one out of a set of three probability models is chosen for the entropy coding of ε based on the value of error strength estimate Δ

$$
\begin{cases}
\text{using} & \text{pmf1}, & \text{if} \quad 0 \le \triangle \le 1 \\
\text{using} & \text{pmf2}, & \text{if} \quad 1 < \triangle \le 55 \\
\text{using} & \text{pmf3}, & \text{otherwise}.
\end{cases} \tag{9}
$$

B. Error Sign Flipping

The concept of error sign flipping in [16], [17], [19] is also used in this paper for coding the refined error signal ε. Since e_p, the amount for prediction error compensation, is an average result of past history or experiences, it is very likely that the refined error ε has the same sign as e_p. Therefore, the refined error ε is encoded according to the following equation:

$$
\begin{cases}
\text{encode} - \varepsilon, & \text{if } e_p < 0 \\
\text{encode } \varepsilon, & \text{otherwise}.
\end{cases} \tag{10}
$$

It is noted that all the pixels used in the proposed coding system are causal, the decoder can calculate the error estimate e_p just like the encoder. Therefore, the decoder can reconstruct the sign flipped errors successfully when the image is to be decompressed.

C. Error Remapping

The range of the refined error ε is $[-255, 255]$. In general, a probability model with a set of 511 symbols should be used for the entropy coding of prediction errors. However, they can only take on values in the range $[-x_{\text{cpd}}, 255 - x_{\text{cpd}}]$ [16], [17], [19]. We will use the following error remapping before it is entropy encoded:

$$
\begin{cases}
\varepsilon = \varepsilon + 256, & \text{if} \quad \varepsilon < -128 \\
\varepsilon = \varepsilon - 256, & \text{if} \quad \varepsilon > 128.
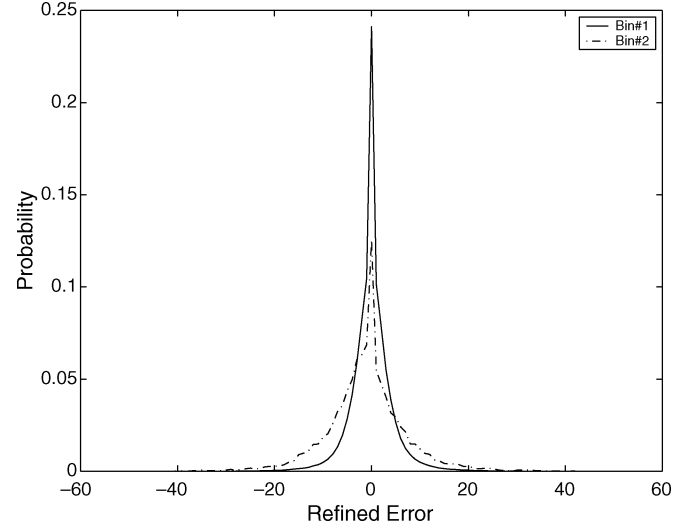\end{cases} \tag{11}
$$

Fig. 5. Histogram of errors in quantization bins for image "Lennagrey." (using a sixth-order LS-based predictor with $\gamma_1 = 100, \gamma_2 = 10$).

In this case, the number of symbols used is reduced to 256, which further improves the coding gain in the entropy coding stage. Since all the pixels used are causal and the decoder performs prediction and compensation just like the encoder, the predicted value x_p, the error estimate e_p and thus the compensated prediction, i.e., $x_{\text{cpd}} = x_p + e_p$, can be calculated in the decoder. Therefore, the decoder can reconstruct the remapped ε by

$$
\begin{cases}
\varepsilon = \varepsilon + 256, & \text{if} \quad \varepsilon < -x_{\text{cpd}} \\
\varepsilon = \varepsilon - 256, & \text{if} \quad \varepsilon > (255 - x_{\text{cpd}}).
\end{cases} \tag{12}
$$

D. Histogram Tail Truncation

With the quantization bins in (9), the error histogram in each quantization bin for image "Lennagrey" (Fig. 4) are plotted in Fig. 5. The curve of Bin3 is not shown because no error strength estimate falls in that region. As can be seen, most of the refined errors fall in the region around 0. Though seldom occur, the count of occurrence for those away from 0 are initialized to be 1 in case they do occur. This operation degrades the performance of entropy coding. To conquer this problem, we use the concept of *histogram tail truncation* in [16].

The probability distribution of the prediction error is usually a two-sided Laplacian distribution [16], [19]. Instead of remapping the error into a one-sided monotonically decreasing probability distribution in [16], the error histogram tail are truncated symmetrically in each quantization bin. The cutoff region for quantization bins are chosen to be $[-25,25]$, $[-48,48]$ and $[-128,127]$ such that over 99% of the refined errors in each quantization bin are within the truncated regions. For example, an error 30 in Bin1, is encoded first to be 25 using pmf1, followed by 5 using pmf2.

VI. Experiments

In this section, we evaluate the performance of the proposed lossless image codec. Comparisons to existing linear and nonlinear predictors and coders are also given. All the test images
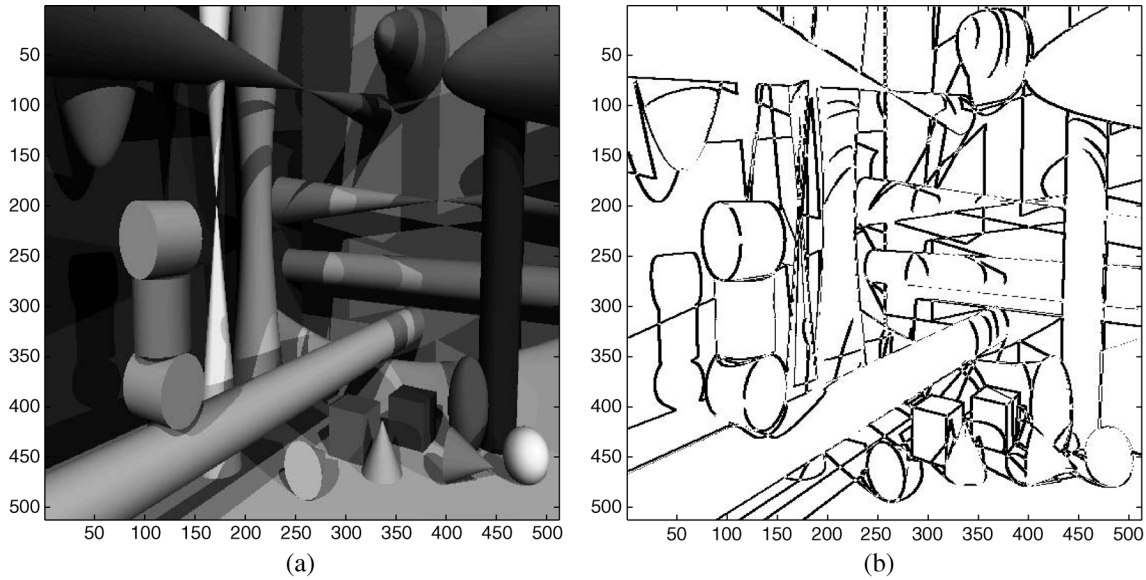
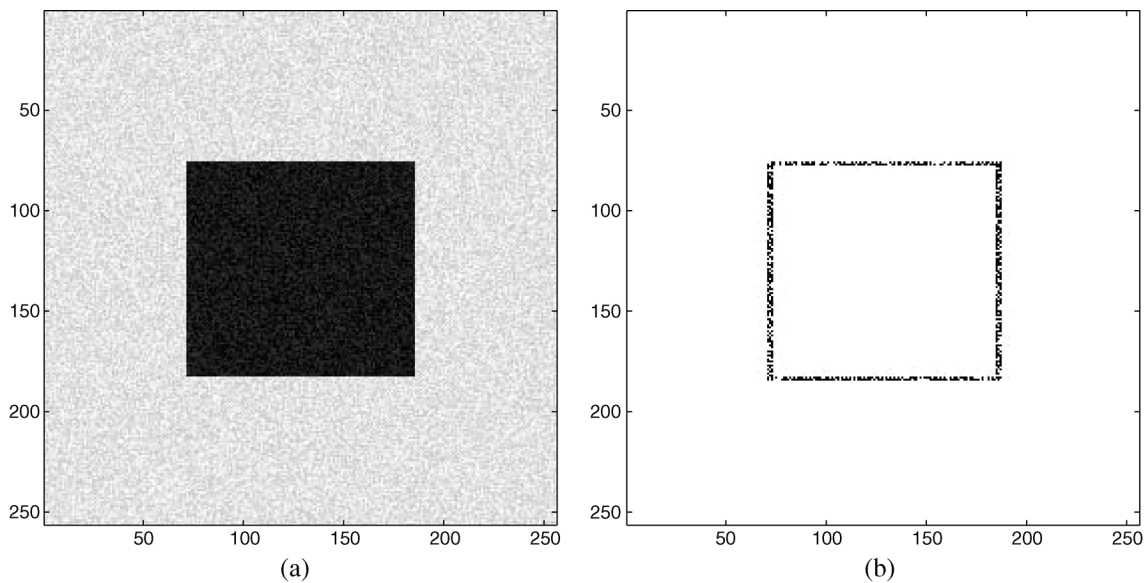Fig. 6. (a) Image "Shapes." (b) Pixels for which (1) is satisfied in the image "Shapes."

Fig. 7. (a) Image "Noisesquare." (b) Pixels for which (1) is satisfied.

used in the experiments are from the website of TMW[1] [20]. We first demonstrate the usefulness of the proposed edge detector and then the error compensation mechanism in the regular mode. Then, the bit rate performance of the system is presented. Finally, the computational complexity of the proposed system will be discussed.

A. Edge Detector

To demonstrate the effectiveness of the proposed edge detector, we use the image "Shapes" [Fig. 6(a)], which is an artificial image with many edges and lines. The pixels that satisfy the two conditions in (1) are marked in Fig. 6(b). We can see from Fig. 6(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"

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

[Fig. 7(a)], an image with salt-and-pepper noise. The pixels picked out by the edge detector are as shown in Fig. 7(b). We see from Fig. 7(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 "Lennagrey," a natural image that is shown in Fig. 4. As can be seen in Fig. 8, the pixels around the edges have been picked out successfully.

B. Error Compensation

To demonstrate the usefulness of the proposed error compensation mechanism in regular mode, we use a sixth-order LS-based predictor. The image "Lennagrey" is used as the test image. The compensated prediction errors for image "Lennagrey" is shown in Fig. 9(a). As can be seen in Fig. 9(a), the proposed approach performs very well around edges and the statistical redundancy has been removed effectively. Fig. 9(b) shows

TABLE I
COMPRESSION RATIO AND THE RUNNING TIME (IN SECONDS, ON A PIII-1.06-GHz MACHINE) OF THE CONSTRUCTED CODER VARY WITH DIFFERENT PREDICTION
ORDER USING THE PROPOSED APPROACH

| Image	N=4		N=6		N=8		N=10	
	Compression Ratio	Run Time	Compression Ratio	Run Time	Compression Ratio	Run Time	Compression Ratio	Run Time
Baboon	1.35	3.82	1.38	6.80	1.38	12.37	1.38	14.21
Lena	1.82	3.09	1.84	4.22	1.84	5.71	1.84	6.95
Lennagrey	2.01	3.04	2.03	3.79	2.03	5.18	2.03	5.84
Peppers	1.86	3.03	1.88	3.86	1.88	4.59	1.89	6.06
Barb	1.88	3.38	1.95	4.80	1.96	9.07	1.97	8.62
Barb2	1.73	5.41	1.77	8.28	1.77	14.33	1.77	15.26
Boats	2.11	4.82	2.15	6.02	2.16	8.43	2.17	9.18
Gold Hill	1.82	4.76	1.83	6.58	1.84	8.30	1.84	10.97
Average	1.82	3.92	1.85	5.54	1.86	8.50	1.86	9.64
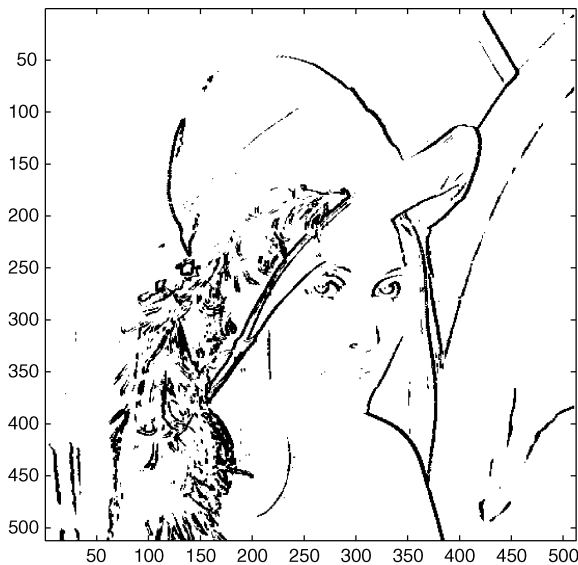
Fig. 8. Pixels for which (1) is satisfied in the image "Lennagrey" ($\gamma_1 = 100$, $\gamma_2 = 10$).

the histograms of prediction error with and without error compensation. We can observe the usefulness of the proposed automatic context modeling for error refinement from Fig. 9(b). The histogram of the refined error has a narrower peak. The first-order entropy for the compensated error is 3.97 bits and 4.20 bits for the uncompensated error.

C. Order of Predictor

The order of the predictor affects coding gain in the regular mode. We list in Table I the compression ratio for order $N = 4,6,8,10$ with the run mode disabled. The execution time is also listed in the Table. As can be seen in Table I, the compression ratio quickly saturates when the prediction order is greater than six. Moreover, the increases in the execution time does not justify the use of prediction order higher than six. Therefore, the use of a sixth-order predictor in the regular mode is a proper choice and this will be used in the design of a lossless image coder afterward for comparison with existing state-of-the-art lossless image coders.

D. Effectiveness of the Edge-Look-Ahead Mechanism

The usefulness of the proposed predictor with edge-look-ahead can be demonstrated through the following experiment. We construct two sixth-order LS based predictors for the regular mode; 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. In this experiment, the run mode is also enabled and the image "Lennagrey" in Fig. 4 is used for this comparison.

For the predictor with edge-look-ahead, the pixels for which LS adaption is activated are shown in Fig. 10(a). Overall, about 17% of pixels activate the LS adaptation process. The image of uncompensated prediction errors $(x_n - x_p)$ and the corresponding histogram are shown in Fig. 10(b) and Fig. 11 respectively. For comparison, we also show in Fig. 11 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 edge-look-ahead approach has made a good tradeoff between prediction efficiency and computational complexity. Indeed, the entropies corresponding to the two histograms in Fig. 11 are, respectively, 4.20 bits (proposed approach) and 4.18 bits (adapted in a pixel-by-pixel manner).

E. Comparisons to Existing Predictors

Table II gives comparisons of uncompensated prediction errors for a set of eight test images in first-order entropies. We compare with existing linear and nonlinear predictors for prediction order $N = 4,6,8,10$. In this experiment, the run mode of the proposed system is disabled; only the regular mode is used so that we can make a fair comparison. The results of a MED [17], a GAP [16] and an edge directed predictor (EDP) with different orders are taken from [26]. As can be seen in Table II, 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 [26]. Moreover, the results of the proposed approach are very close to those with pixel-by-pixel LS adaptation.
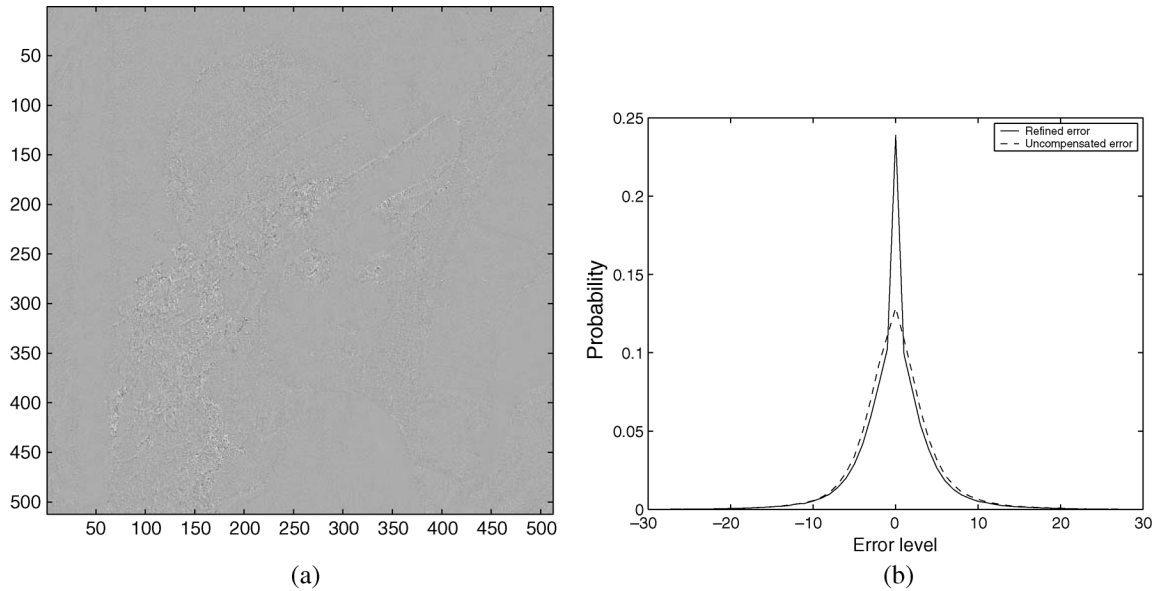
Fig. 9. (a) Image of refined errors using the proposed approach for "Lennagrey." (using a sixth-order LS-based predictor with $\gamma_1 = 100, \gamma_2 = 10$) (b) histogram of prediction errors for image "Lennagrey."
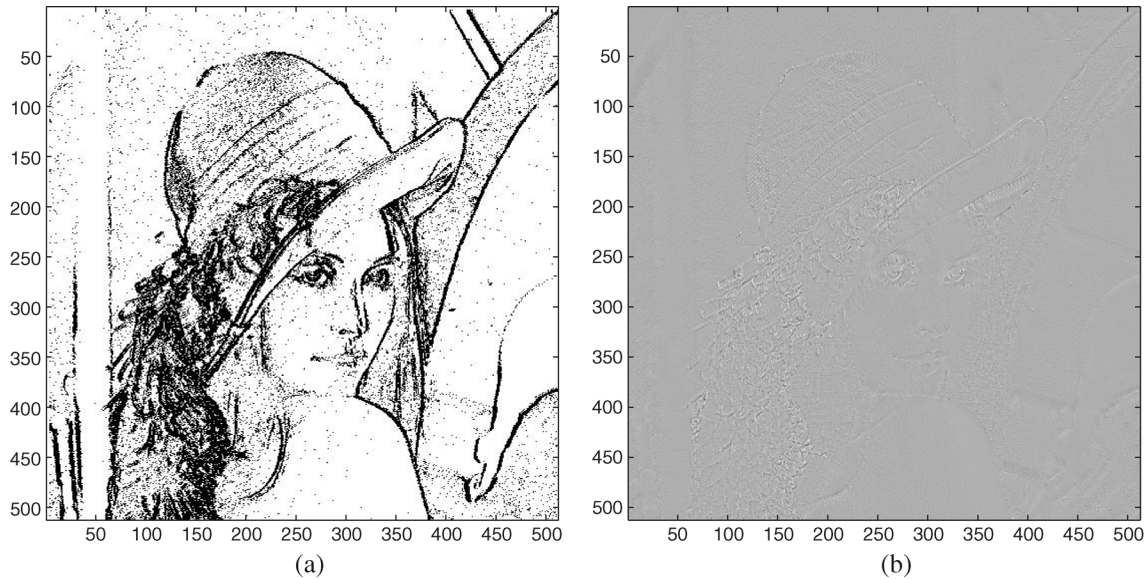

Fig. 10. (a) Pixels for which LS adaption is used in the proposed edge-look-ahead predictor for the image "Lennagrey." (using a sixth-order predictor with $\gamma_1 = 100, \gamma_2 = 10$) (b) image of uncompensated prediction errors using the proposed edge-look-ahead approach for "Lennagrey."

F. LS Adaptation

In this subsection, we investigate how the prediction performance (entropy) varies with the variance threshold γ_1 in (1) for LS adaptation. We construct a tenth-order predictor with edge-look-ahead for the experiment and the image "Lennagrey" (Fig. 4) is used for the test. For LS adaptation, we use the same training area as defined in EDP [26]. Moreover, we set $\gamma_2 = 10$ and $\theta = 10$ for all cases in the experiment. By varying the variance threshold γ_1, the number of pixels that activates LS adaptation also changes. The experimental results using the proposed approach with various γ_1 are shown in Table III.

We observe from Table III that a small γ_1 may results in a small entropy at the expense of an increased number of pixels performing LS adaptation process. The percentage of pixels regarded as around an edge increases as γ_1 decreases, but the

percentage of pixels that activates the LS adaptation in slowly varying areas almost remain unchanged. This can be best observed from the fifth column, which is obtained by subtracting the fourth column from the third column, of Table III, and we can find that about 3% to 5% of pixels in slowly varying areas will activates the LS adaptation. Therefore, the improvement on the entropy in the proposed approach is mainly around edges.

G. Further Insight on the Proposed Edge-Look-Ahead Mechanism

To highlight the usefulness of the proposed edge-look-ahead mechanism for LS-based predictor, we compare the proposed approach and the state-of-the-art LS-based EDP predictor in [26]. As in [26], we use a tenth-order predictor. With EDP [26], the percentage of pixels activating LS adaptation is 9.87% and the resulting entropy is 4.22 bpp. The results of the proposed

TABLE II
FIRST-ORDER ENTROPIES OF PREDICTION ERRORS. (ONLY THE REGULAR MODE IS USED IN THE PROPOSED ALGORITHM; THE RUN MODE IS DISABLED)

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 III
PERCENTAGE OF PIXELS PERFORMING LS ADAPTION AND THE RESULTING FIRST-ORDER ENTROPY BY VARYING THE VARIANCE THRESHOLD γ_1 IN THE PROPOSED APPROACH (THE IMAGE "LENNAGREY" IS USED FOR THE TEST WITH $\gamma_2 = 10$, $\theta = 10$ FOR ALL CASES)

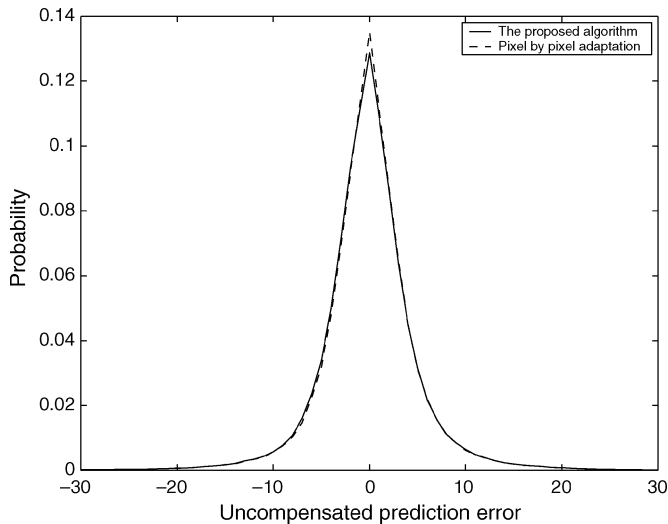| Parameter	Number of Pixels with LS Adaptation	Percentage with LS Adaptation (A)	Percentage Detected as around an Edge (B)	Percentage with LS Adaptation in slowly varying areas (C=A-B)	First-order Entropy
$\gamma_1=100$	37692	14.38	11.47	2.91	4.176
$\gamma_1=200$	28205	10.76	6.87	3.89	4.184
$\gamma_1=300$	24117	9.20	4.79	4.41	4.188
$\gamma_1=400$	21819	8.32	3.58	4.75	4.190
$\gamma_1=500$	20342	7.76	2.75	5.01	4.193

Fig. 11. Histogram of uncompensated prediction errors for the proposed approach and that of a pixel-by-pixel adaptation. (both using a sixth-order predictor).

tion error if EDP is used for those pixels instead. As can be seen in Fig. 12(a), the histogram with the proposed approach is much narrower than that with EDP. The proposed approach has a smaller prediction error than EDP does around edges. Indeed, the entropies corresponding to the two histograms in Fig. 12(a) are, respectively, 5.12 bits (proposed) and 5.32 bits (EDP).

As another example, we use the artificial image "Shapes" [Fig. 6(a)] for the same experiment. The pixels for which (1) is satisfied have been shown in Fig. 6(b). The histogram of uncompensated prediction errors using the proposed approach for those pixels in Fig. 6(b) is shown in Fig. 12(b). The histogram with the proposed approach is much narrower than that with EDP; the proposed approach has a smaller prediction error than EDP does around edges. The entropies corresponding to the two histograms in Fig. 12(b) are, respectively, 4.51 bits (proposed) and 5.28 bits (EDP).

As can be seen in Fig. 12(a) and (b), the proposed system has achieved a noticeable improvement over the use of EDP around edges. Moreover, the improvement is very distinct for images with many edges and lines. As indicated in the title of EDP [26], the EDP is designed mainly for natural images and it is noted that most of the areas vary slowly in natural images. The EDP initiates the LS adaptation process only after the prediction error is beyond a preselected threshold so that the computational complexity can be reduced. Nevertheless, an abrupt change in the image pixel, e.g., an edge or a line, may results in a large prediction error with EDP. Compared with EDP, the proposed system can look ahead to 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. With a moderately increased computational complexity in detecting

edge-look-ahead approach are given in Table III. As can be seen in the rows for which γ_1 varies from 300 to 500 (the last three rows of Table III), the number of pixels performing LS adaptation is fewer than that with EDP approach but still have lower entropy than EDP.

To gain a further insight, we look into the case with $\gamma_1 = 100$. The histogram of uncompensated prediction errors for those pixels that are considered as around an edge [i.e., pixels for which (1) are satisfied] is shown in Fig. 12(a). For comparison, we also show the histogram of uncompensated predic-
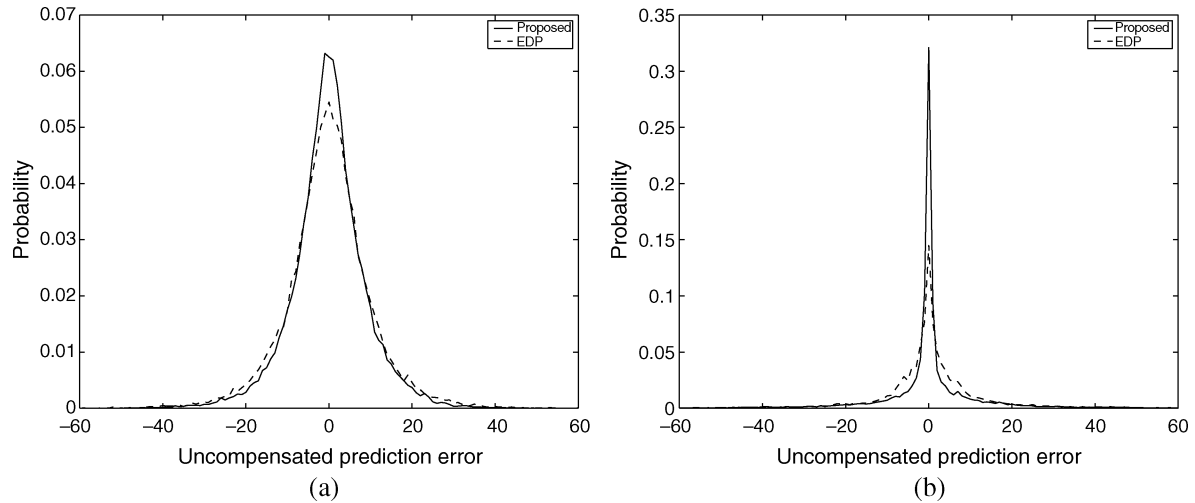
Fig. 12. (a) Histogram of uncompensated prediction errors for the pixels in Fig. 8. (b) Histogram of uncompensated prediction errors for those pixels in Fig. 6(b).

TABLE IV
COMPARISONS WITH EXISTING LOSSLESS IMAGE CODERS (IN BITS/SAMPLE). THE FIFTH COLUMN IS THE EXECUTION TIME OF
THE PROPOSED APPROACH ON A PIII-1.06-GHz MACHINE

Image	First-oreder Entropy	RALP	% of LS adaptation	seconds	JPEG-LS [17]	CALIC [16]	EDP* [26]	TMW [20]
Airplane	3.82	3.71	16.6	3.75	3.82	3.74	N/A	3.60
Baboon	5.91	5.81	63.8	6.82	6.04	5.88	5.81	5.73
Balloon	2.56	2.55	4.0	4.50	2.90	2.83	N/A	2.66
Barb	4.15	4.12	33.6	4.80	4.69	4.32	4.11	4.09
Barb2	4.57	4.51	37.7	8.02	4.69	4.53	4.52	4.38
Boats	3.84	3.75	18.1	6.07	3.93	3.83	3.80	3.61
Camera	4.42	4.24	26.7	1.18	4.31	4.19	N/A	4.10
Couple	3.63	3.63	15.8	1.01	3.70	3.61	N/A	3.45
Gold Hill	4.41	4.32	23.6	6.81	4.48	4.39	4.39	4.27
Lena	4.36	4.35	23.5	4.27	4.61	4.48	4.40	4.30
Lennagrey	3.97	3.95	17.4	3.82	4.24	4.11	4.02	3.91
Noisesquare	5.35	5.37	56.7	1.60	5.68	5.44	N/A	5.54
Peppers	4.27	4.27	18.2	3.97	4.51	4.42	4.35	4.25
Shapes	1.87	1.52	7.1	1.94	1.21	1.14	N/A	0.76
Average	4.08	4.01	25.92	4.18	4.20	4.07	4.43	3.90

(* For the EDP method, the bit rates for some of the images are not available and the average
is computed only for those that are available.)

(* For the EDP method, the bit rates for some of the images are not available and the average
is computed only for those that are available.)

the existence of an edge, the proposed system has achieved a noticeable improvement around edges than that with the EDP approach.

H. Comparisons to Existing State-of-the-Art Coders

Table IV gives the actual bit rates of proposed RALP coder, JPEG-LS [17], CALIC [16], EDP [26] and TMW [20] for a set of fourteen test images. In Table IV, the results of CALIC, EDP and TMW are taken directly from [21] and those of the JPEG-LS are obtained using the code given in the website of LOCO-I [17]. All the bit rates of the proposed algorithm are obtained using the same parameters described in previous sections and no individual optimization is performed. Besides, we show in the second column and the fourth column, respectively, the

first-order entropies of the compensated prediction errors and the percentage of pixels performing LS adaptation using the proposed approach. Moreover, we also show in the fifth column the execution time of the proposed coder so that we can get a picture on the runtime performance of the proposed approach. It should be noted that some of the results in EDP are denoted by "N/A" because they are not reported in the paper of EDP [26] and [21]. Therefore, the set of images included in the average is different for that column than any other. Table IV shows that RALP has lower bit rates than JPEG-LS in thirteen out of the fourteen test images and outperforms CALIC [16] in eleven of fourteen test images. Encouragingly, the proposed RALP achieves lower bit rates than the highly complex TMW in two images, "Balloon" and "Noise square." It should be noted that the proposed bit rate

TABLE V
OPERATION COUNTS FOR EDGE DETECTOR IN (1)

Operation	Compare	ADD/SUB	MUL/DIV	Square
Edge detection	$\leq$ (N+2)	$\leq$ (4N-2)	$\leq$ 7	$\leq$ (N+3)
"N" is the number of pixels used in edge detection. In this paper, N= 4.				

performance for the artificial image "Shapes" is inferior to that of obtained by CALIC [16] and JPEG-LS [17]. For this, we can see from Fig. 6(a) that the image "Shapes" can be segmented into many slowly varying areas in which the run-length coding is most efficient to be used. Therefore, the bit rate performance for the artificial image "Shapes" can be improved if a more sophisticated run-length encoding technique is used in the proposed approach.

I. Computational Complexity

Numerically, the normal equations (4) can be solved by *Cholesky decomposition* or *SVD* depending on the rank of $\mathbf{P}$ in (3). When $\mathbf{P}$ is full-ranked, the *Cholesky decomposition* can be used and it requires only $N^3/6$ multiplications to solve (4), which is about half the usual number of multiplications required by alternative methods [33], [34]. Only when $\mathbf{P}$ is not full ranked, *SVD*, which requires much higher computations, is needed. Fortunately, our experiments show that in most cases $\mathbf{P}$ has full rank. This is because pixels around boundaries usually have a large variation in gray levels and thus the matrix $\mathbf{P}$ in (3) is seldom rank deficient. Therefore, most of the computations take place in forming the normal equations (4) rather than solving them. For this, an inclusion and exclusion method for fast construction of the $\mathbf{P}^T\mathbf{P}$ matrix has been proposed in [28].

For the proposed edge detector, the operation counts for each coding pixel in the edge detection process are listed in Table V. It should be noted that there is no need to check both of the two inequalities in (1) for every pixel. Only when the first inequality holds then we check the second condition. Therefore, the actual computational cost is lower than what is listed in Table V. Though edge detection incurs a slight increase in computations, the overall complexity is reduced significantly when compared with that of pixel-by-pixel adaptation approach. The proposed approach has achieved a very good trade-off between runtime performance and prediction efficiency.

VII. CONCLUSION

In this paper, a switching coding scheme that combines the advantages of both RALP coding is proposed. For pixels in slowly varying areas, run-length coding is used; otherwise an LS-based adaptive predictor is used. Instead of performing LS adaptation in a pixel-by-pixel manner, we adapt the predictor coefficients only when an edge is detected or when the prediction error is greater than a predefined threshold so that the computational complexity can be significantly reduced. For this, we use a simple yet effective edge detector using only causal pixels. This way, the proposed system can look ahead to determine if the coding pixel is around an edge and initiate the LS adaptation in advance to prevent the occurrence of a large prediction error.

With the proposed switching structure, very good prediction results can be obtained in both slowly varying areas and pixels around boundaries. Moreover, only causal pixels are used for estimating the coding pixels in the proposed encoder; no additional side information needs to be transmitted. 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 prediction results has been obtained. Furthermore, comparisons to existing state-of-the-art lossless image predictors and coders have demonstrated the superiority of the proposed system.

REFERENCES

[1] K. Sayood, *Introduction to Data Compression*, 2nd ed. New York: Morgan-Kaufmann, 2000, pp. 139–163.

[2] K. R. Rao and J. J. Hwang, *Techniques & Standards for Image, Video & Audio Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1996, pp. 31–42.

[3] B. Carpentieri, M. J. Weinberger, and G. Seroussi, "Lossless compression of continuous-tone images," *Proc. IEEE*, vol. 88, no. 11, pp. 1797–1809, Nov. 2000.

[4] N. Memon and X. Wu, "Recent developments in context-based predictive techniques for lossless image compression," *Comput. J.*, vol. 40, no. 2/3, pp. 127–136, 1997.

[5] X. Wu, "An algorithmic study on lossless image compression," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar.-Apr. 1996, pp. 150–159.

[6] X. Wu and T. Qiu, "Wavelet coding of volumetric medical images for high throughput and operability," *IEEE Trans. Med. Imag.*, vol. 24, no. 6, pp. 719–727, Jun. 2005.

[7] A. T. Deever and S. S. Hemami, "Lossless image compression with projection-based and adaptive reversible integer wavelet transforms," *IEEE Trans. Image Process.*, vol. 12, no. 5, pp. 489–499, May 2003.

[8] M. Grangetto, E. Magli, M. Martina, and G. Olmo, "Optimization and implementation of the integer wavelet transform for image coding," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 596–604, Jun. 2002.

[9] N. V. Boulgouris, D. Tzovaras, and M. G. Strintzis, "Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 1–14, Jan. 2001.

[10] J. Reichel, G. Menegaz, M. J. Nadenau, and M. Kunt, "Integer wavelet transform for embedded lossy to lossless image compression," *IEEE Trans. Image Process.*, vol. 10, no. 3, pp. 383–392, Mar. 2001.

[11] 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.

[12] M. K. Das, C. C. Li, and S. R. Burgett, "A new multiresolution predictive scheme for lossless compression of medical images," in *Proc. IEEE Int. Symp. Computer-Based Medical Systems*, Jun. 13–16, 1993, pp. 34–39.

[13] N. Memon, X. Kong, and J. Cinkler, "Context-based lossless and near-lossless compression of EEG signals," *IEEE Trans. Info. Technol. Biomed.*, vol. 3, no. 9, pp. 231–238, Sep. 1999.

[14] K. H. Yang and A. F. Faryar, "A contex-based predictive coder for lossless and near-lossless compression of video," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 10–13, 2000, vol. 1, pp. 144–147.

[15] X. Y. Lin, L.-K. Shark, M. R. Varley, and B. J. Matuszewski, "A hybrid lossless compression scheme using region-based predictive coding and integer wavelet transform," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 7–10, 2001, vol. 3, pp. 486–489.

[16] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[17] 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 [Online]. Available: http://www.hpl.hp.com/loco/locodownold.htm

[18] G. Motta, J. A. Storer, and B. Carpentieri, "Lossless image coding via adaptive linear prediction and classification," *Proc. IEEE*, vol. 88, no. 11, pp. 1790–1796, Nov. 2000.

[19] 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.

[20] B. Meyer and P. E. Tischer, "TMW-a new method for lossless image compression," in *Proc. Int. Picture Coding Symp.*, Berlin, Germany, Oct. 1997 [Online]. Available: http://www.csse.monash.edu.au/~bmeyer/tmw/paper.ps

[21] B. Aiazzi, L. Alparone, and S. Baronti, "Fuzzy logic-based matching pursuits for lossless predictive coding of still images," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 4, pp. 473–483, Aug. 2002.

[22] L.-J. Kau, Y.-P. Lin, and C.-T. Lin, "Lossless image coding using adaptive, switching algorithm with automatic fuzzy context modeling," *Proc. IEE Vision, Image, Signal Process.*, vol. 153, no. 5, pp. 684–694, Oct. 2006.

[23] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, no. 2, pp. 288–303, Feb. 1995.

[24] E. D. Sontag, "Feedback stabilization using two-hidden-layer nets," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 981–990, Nov. 1992.

[25] C.-T. Lin and C. S. G. Lee, *A Neural-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996, pp. 235–250.

[26] X. Li 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.

[27] L.-J. Kau and Y.-P. Lin, "Adaptive lossless image coding using least-squares optimization with edge-look-ahead," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 11, pp. 751–755, Nov. 2005.

[28] N. Kuroki, T. Nomura, M. Tomita, and K. Hirano, "Lossless image compression by two-dimensional linear prediction with variable coefficients," *IEICE Trans. Fundam.*, vol. E75-A, no. 7, pp. 882–889, Jul. 1992.

[29] 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.

[30] B. Meyer and P. E. Tischer, "Glicbawls—Grey level image compression by adaptive weighted least squares," in *Proc. Data Compression Conf.*, Snowbird, Utah, 2001, p. 503.

[31] H. Ye, G. Deng, and J. C. Devlin, "Adaptive linear prediction for lossless coding of greyscale images," *Proc. IEEE Image Process.*, vol. 1, pp. 128–131, Sep. 2000.

[32] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[33] S. J. Leon, *Linear Algebra With Applications*. Englewood Cliffs, NJ: Prentice-Hall, 2002, pp. 482–488.

[34] 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 [Online]. Available: http://www.library.cornell.edu/nr/cbookcpdf.html

Lih-Jen Kau (S'05–M'07) was born in Hualien, Taiwan, R.O.C., in 1969. He received the B.S. degree in control engineering and the M.S. degree in electrical and control engineering both from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 1991 and 1997, respectively. He is currently working toward the Ph.D. degree in electrical and control engineering at NCTU.

From 1996 to 1998, he was with the Chunghwa Telecom, Taipei, Taiwan, R.O.C., where he served as a Senior Technician. In August 1998, he joined the Department of Computer and Communication Engineering, Dahan Institute of Technology, Hualien, Taiwan, R.O.C., as an instructor, and since February 2006, he has been an Assistant Professor and the Director of Computer Center in Dahan. His research interests include signal processing, microprocessors and digital/analog circuits design.

Dr. Kau is a recipient of the 2003 ZyXEL scholarship Award, ZyXEL Comm. Corp., Hsinchu, Taiwan, R.O.C.

Yuan-Pei Lin (S'93–M'97–SM'03) was born in Taipei, Taiwan, R.O.C., in 1970. She received the B.S. degree in control engineering from the National Chiao-Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 1992, and the M.S. and Ph.D. degrees, both in electrical engineering from California Institute of Technology, Pasadena, in 1993 and 1997, respectively.

She joined the Department of Electrical and Control Engineering of NCTU in 1997. Her research interests include digital signal processing, multirate filter banks, and signal processing for digital communication, particularly the area of multicarrier transmission.

Dr. Lin is a recipient of 2004 Ta-You Wu Memorial Award. She served as an Associate Editor of IEEE TRANSACTION ON SIGNAL PROCESSING 2002–2006. She is currently an Associate Editor of IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTION ON CIRCUITS AND SYSTEMS II, *EURASIP Journal on Applied Signal Processing*, and *Multidimensional Systems and Signal Processing*. She is also a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2006–2007.