# Lossless image coding using adaptive, switching algorithm with automatic fuzzy context modelling

L.-J. Kau, Y.-P. Lin and C.-T. Lin

**Abstract:** A switching adaptive predictor (SWAP) with automatic fuzzy context modelling is proposed for lossless image coding. Depending on the context of the coding pixel, the SWAP encoder switches between two predictors: the adaptive neural predictor (ANP) and the texture context matching (TCM) predictor. The ANP is known to perform well and gives small prediction errors except for pixels around edges. For areas with edges, TCM is used. To decide which is to be used, a switching criterion is proposed to pick out pixels around edges effectively. With the switching predictor structure, small prediction errors can be achieved in both slowly varying areas and edges. Furthermore, the use of the so-called fuzzy context clustering for prediction error refinement is proposed. The proposed compensation mechanism is proved to be very useful through experiments. It further improves the bit rates by, on average, 0.2 bpp in test images. The experiments also show that an average improvement of 0.3 and 0.05 bpp in first-order entropy can be achieved when the proposed switching predictor is compared with the gradient adjusted predictor and a six-order edge directed predictor, respectively. Moreover, the lossless image coder built upon the proposed algorithm also provides lower bit rates than the state-of-the-art context-based, adaptive, lossless image coding (CALIC) system and is comparable to that obtained by the highly complex two-pass coder called TMW.

## 1 Introduction

There have been great advances in lossless image coding recently [1–33]. Some of them are based on the 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 those 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 framework has been proposed [11]. The results presented by Avcibas *et al.* [11] are the equal of those obtained by state-of-the-art compression schemes.

Most of the impressive results are obtained based on predictive coding (as shown in Fig. 1) with context modelling [11–33]. To accommodate varying statistics of coding images, adaptive predictors are often used [12–32]. In [12–20], adaptive prediction is achieved by using multi-predictor structures. The context-based, adaptive, lossless image coding (CALIC) system [16], a state-of-the-art lossless predictor proposed for JPEG-LS, applies a gradient adjusted predictor (GAP). On the basis of the gradient of neighbouring pixels, one out of a set of seven predictors is chosen. The LOCO-I coder [17, 18], an algorithm motivated by CALIC [16], uses a median edge detector (MED) to choose one of the three predictors for current prediction. The LOCO-I system has been standardised into JPEG-LS. In adaptive linear prediction and classification (ALPC) [19], predictor coefficients are adapted in the coding process by applying the gradient descent rule. By Yu [20], the prediction value of the coding pixel is the weighted sum of the outputs of five predictors. Recently, the least squares (LS) optimisation has been noted as an efficient approach for the adaption of predictor coefficients [21–26]. However, a pixel-by-pixel adaptation of predictor coefficients is regarded as prohibitive. Therefore some of the results have been proposed for the reduction of computational complexity during the prediction process [22, 23].

Multiple-pass prediction is introduced by Wu [27] and Meyer and Tischer [28]. With multiple passes, the encoder can form a 360° prediction [27] or perform a global image analysis [28]. A highly complex two-pass coder called TMW has been proposed by Meyer and Tischer [28]. Using multiple linear predictors and global image analysis, the TMW system can achieve lower bit rates than the existing coders for most images. Slyz and Neuhoff [33] proposed the idea of estimating coding pixel on the basis of context matching. For each coding pixel, a causal area of dimensions $30 \times 30$ is used for context matching. From this area, 11 pixels are chosen and averaged to form an estimate of the coding pixel. The histogram of the prediction errors corresponding to the 11 candidates is calculated. By computing the variance of the 11 prediction errors, one out of a set of 37 Laplacian distributions is chosen. The prediction error of the coding pixel is then coded using a conditional arithmetic coder corresponding to the chosen Laplacian model. Wu and Memon [16] demonstrated that prediction error can be further refined through error compensation. The compensated error has a



*IEE Proceedings* online no. 20045256

doi:10.1049/ip-vis:20045256

Paper first received 5th December 2004 and in revised form 12th December 2005

The authors are with the Department of Electrical and Control Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan, Republic of China

E-mail: u8912801@cc.nctu.edu.tw

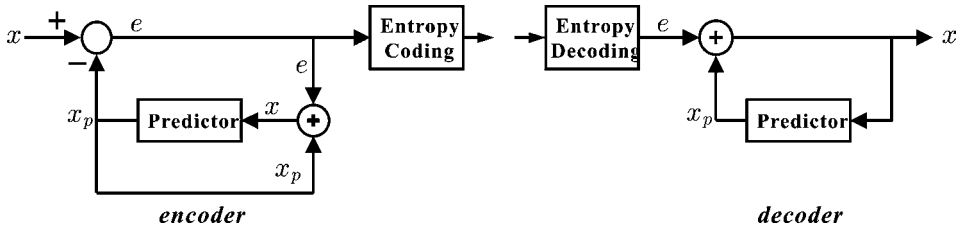


**Fig. 1** *Structure of lossless predictive codec*

narrower histogram and hence a lower first-order entropy. In CALIC [16], 576 compound contexts are used for error modelling.

In the context of optimal predictors, the minimum mean square error estimate of $Y$ given observations $X_1, X_2, \ldots, X_n$ is $E\{Y|X_1, X_2, \ldots, X_n\}$, and is generally a non-linear function. There have been many results using neural networks as non-linear estimators [29–32]. Dony and Haykin [29] proposed neural approaches to predictive image compression. Neural predictors based on multi-layered perceptrons are used by Manikopoulos [30], Li [31] and Marusic and Deng [32]. A functional-link neural network predictor is proposed for low-complexity coding by Manikopoulos [30]. Li [31] and Marusic and Deng [32] proposed updating the network weights using the prediction errors of coded pixels to make the predictor adaptive. Non-linear predictors using neural networks, though performing well in slowly varying areas [34], can have large prediction errors around edges [32, 35]. The result can be improved by using additional hidden layers or hidden neurons but this can incur a drastic increase in complexity [36].

In this paper, we propose a prediction scheme for lossless image coding, called the switching adaptive predictor (SWAP), composed of four components as shown in Fig. 2. The proposed SWAP coder switches between two predictors: the adaptive neural predictor (ANP) and the texture context matching (TCM) predictor. The ANP is an adaptive three-layered back propagation network that is updated on-the-fly using causal pixels as training patterns. It is known to perform well and gives small prediction errors except for pixels around edges, for which TCM is used instead. When TCM is used, pixels in a predefined causal area having textures similar to that of the coding pixel are averaged for the current prediction. We will see that the TCM provides a good complement to ANP. Very good predictions can be obtained for pixels around edges where conventional predictors tend to have large prediction errors. To determine whether the coding pixel is around an edge, we also propose a predictor switch (Fig. 2) that uses only causal pixels. With the proposed edge detector, the encoder switches automatically between ANP and TCM. The proposed predictor switch, though very simple, can pick out the edges successfully in the experiments.

It is known that the prediction can be further refined through error compensation. For this, automatic error modelling is achieved in SWAP using the proposed fuzzy context clustering, which is a modified version of the unsupervised fuzzy competitive learning (UFCL) algorithm proposed by Looney [37]. It does not require training data to be available all at once and it can be used for sequential encoding. Furthermore, the number of clusters may vary according to the input image. The output $x_p$ from ANP or TCM is adjusted by the addition of $e_p$, the compensation that is determined through the proposed fuzzy context clustering, to obtain a refined prediction $x_{cpd} = x_p + e_p$. The compensated error signal $\varepsilon = x - x_{cpd}$ can then be entropy-encoded using conditional arithmetic coding [38].

### 1.1 Novelty of the SWAP system

We propose a novel switching mechanism that switches between two predictors: ANP and TCM. The ANP, thoroughly studied in the literature, is known to provide good prediction of pixels in slowly varying areas. For pixels around edges, where the prediction error of ANP tends to be large, the TCM predictor is switched on. Examples are given to demonstrate that TCM predictor renders a good complement to ANP. A simple, yet effective, switching algorithm using only causal pixels is proposed to detect edges so that the decoder knows exactly which predictor is used for each pixel. For error refinement, automatic error modelling is achieved in SWAP using the proposed fuzzy context modelling which leads to a modelling of errors that adapts itself to the input statistics. The combination of a switching structure and an automatic error modelling renders the proposed SWAP coder highly adaptable and very low bit rates can be achieved.

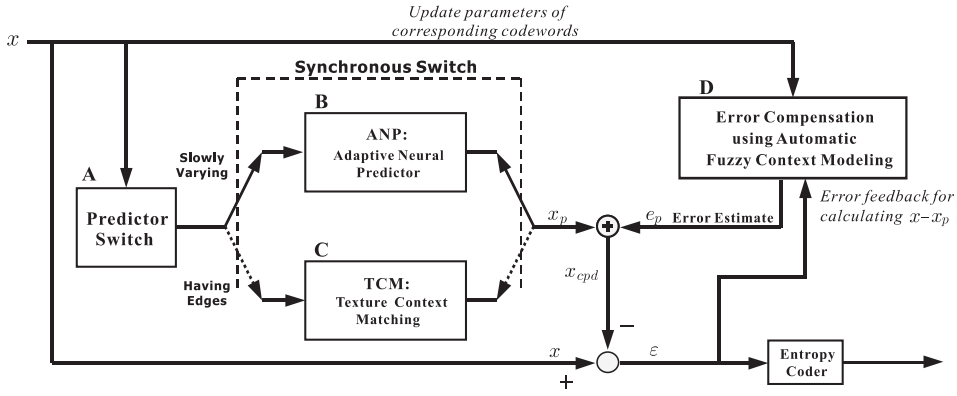


**Fig. 2** *Proposed SWAP system*

## 2 Predictor switch

This section introduces the switching criterions of the proposed SWAP system. The TCM predictor is used in regions with edges whereas ANP is used in all other cases. To determine whether the coding pixel is around an edge, we propose a very simple algorithm that uses only causal pixels, that is, pixels already coded earlier. As much of the systems encode pixels in a raster scan order, it should be noted that conventional edge detectors, for example 'Sobel' operator, cannot 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 whether ANP or TCM should be used. We define the texture context $\kappa$ of a coding pixel as the collection of the ten causal pixels $x_1, x_2, \ldots, x_{10}$ (Fig. 3)

$$\kappa = \{x_1, x_2, \ldots, x_{10}\} \tag{1}$$

The mean $\bar{x}$ and variance $\sigma^2$ of the texture context are calculated and the ten pixels can be divided into two groups, the pixels with grey levels higher than $\bar{x}$ in one group, $\kappa_h$, and the rest in another, $\kappa_l$. We also compute the variance $(\sigma_h^2, \sigma_l^2)$ of those pixels in $\kappa_h$ and $\kappa_l$, respectively.

A histogram with edges is likely to have a large $\sigma^2$ but small $\sigma_h^2$ and $\sigma_l^2$. We determine whether the coding pixel is around an edge if the following two conditions are satisfied

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

where 0.01 is added so that the denominator of (2) does not become 0 when $\sigma_h^2$ and $\sigma_l^2$ are both zero. The case for $\sigma_h^2$ and $\sigma_l^2$ are both zero can occur in an artificial image or when both the members in $\kappa_l$ and $\kappa_h$ have identical grey values. It is noted that the second condition in (2) is included because a region with uniformly distributed grey values also results in a large $\sigma^2$. Therefore the switch first examines if $\sigma^2 \geq \gamma_1$ when a large $\sigma^2$ is detected; then the switch checks the second inequality in (2). The TCM predictor is used whenever the two conditions in (2) are satisfied; otherwise the ANP is used. 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. Moreover, the above switching algorithm is summarised in the following pseudo code.

**Pseudo code for the predictor switch**

```
Calculate the mean x̄ and the variance σ² of the ten pixels in
texture context κ;
  if σ² ≥ γ1
  {
     for (i = 1; i < 11; i++)
     {
        if (xi > x̄)
           xi is added to group κh;
        else
           xi is added to group κl;
     }
     Calculate the variance σh² and σl² of κh and κl;
     if σ²/(0.01 + σh² + σl²) ≥ γ2
           use TCM predictor; /* the case that an edge exists */
     else
           use ANP;
  }
  else
           use ANP;
```

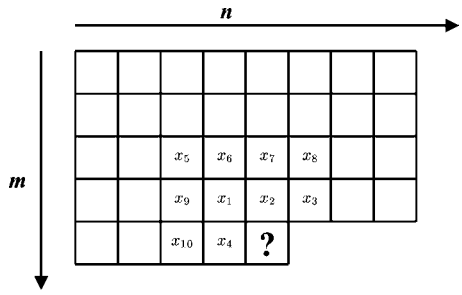


**Fig. 3** *Ten pixels in the texture context of the coding pixel*

## 3 Adaptive neural predictor

The ANP is based on a three-layered back propagation neural network as shown in Fig. 4. There are ten neurons in the input layer, five neurons in the hidden layer and one neuron in the output layer. The output is the prediction value of the coding pixel. The number of hidden neurons is chosen empirically. We have found five to be a proper choice. Our experiments show that, increasing the number of hidden neurons leads to only marginal improvement in entropy or bit rate, but will increase complexity dramatically. We use the ten causal pixels $x_1, x_2, \ldots, x_{10}$ (Fig. 3) in texture context (1) as the predictor inputs. The 36 causal neighbours shown in Fig. 5 are used for training. The ANP adapts itself to the varying statistics by learning from prediction errors of the training patterns. The gradient descent method is used to update network weights continuously on-the-fly.

The neurons in ANP are made up of perceptrons of the same structure. The integration function $f(\cdot)$ is given by

$$f = \sum_{j=1}^{N} w_j y_j \tag{3}$$

where $N$ is the number of prediction inputs, $y_j$ is the input to the neuron and $w_j$ is the weight corresponding to input $y_j$. We use the unipolar sigmoid function $a(\cdot)$ as the activation function

$$Y = a(f) = \frac{1}{1 + e^{-f}} \tag{4}$$

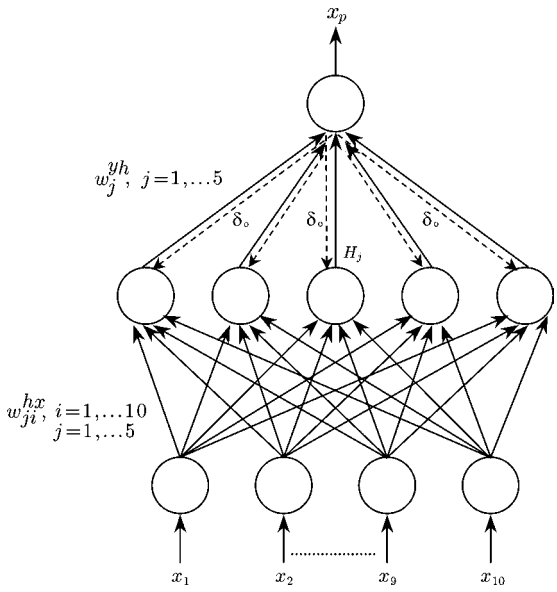


**Fig. 4** *Adaptive nonlinear predictor*

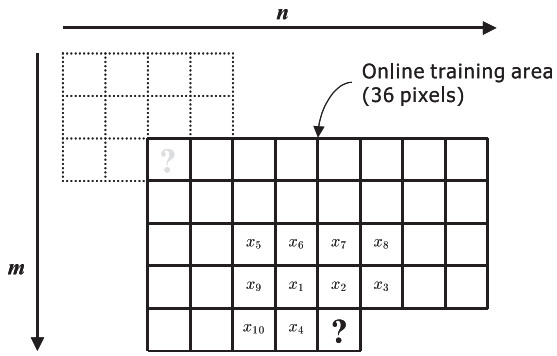


**Fig. 5** *Online training regions for ANP*

where $f$ is as given in (3). The output of the unipolar sigmoid function is bounded between 0 and 1. Suppose the input image has 8 bpp. The pixel values range between 0 and 255. We normalise the pixel values by 255 and use the normalised values as the inputs of the neural network. The output of the network is scaled by 255 and then rounded to the nearest integer.

### 3.1 Weight training

The connection weights $w_j$ of the network are updated using the gradient descent method. The dotted lines shown in Fig. 4 mean that the error signal $\delta_o$ of the output layer is backward propagated to the hidden and input layers for updating current weights. As we are using the unipolar sigmoid function as the activation function and we have only one neuron in the output layer, the error signal $\delta_o$ of the output layer [36] is

$$\delta_o = x_p(1 - x_p)(x - x_p) \tag{5}$$

where $x$ in (5) is the desired output, and $x_p$ is the actual output of the network. On the other hand, the error signal $\delta_{hj}$ of the $j$th neuron in the hidden layer [36] is

$$\delta_{hj} = H_j(1 - H_j)w_j^{yh}\delta_o, \quad j = 1, \ldots, 5 \tag{6}$$

where $H_j$ denotes the output of the $j$th neuron in the hidden layer and $w_j^{yh}$ is the weight connecting the $j$th neuron in the hidden layer to the output neuron. The weight connecting the $j$th neuron in the hidden layer to the output neuron is updated according to Lin and Lee [36]

$$w_j^{yh} = w_j^{yh} + \Delta w_j^{yh}, \quad j = 1, \ldots, 5 \tag{7}$$

The weight increment $\Delta w_j^{yh}$ is calculated according to Lin and Lee [36]

$$\Delta w_j^{yh} = \eta\delta_o H_j + \alpha\Delta w_j^{yh}, \quad j = 1, \ldots, 5 \tag{8}$$

where $\eta$ is the learning rate, and $\alpha$ is a momentum term to accelerate convergence speed [36]. The weight connecting the $i$th input neuron to the $j$th hidden neuron is calculated as follows [36]

$$w_{ji}^{hx} = w_{ji}^{hx} + \Delta w_{ji}^{hx}, \quad \text{with} \quad \Delta w_{ji}^{hx} = \eta\delta_{hj}x_i + \alpha\Delta w_{ji}^{hx}, \quad j = 1, \ldots, 5 \text{ and } i = 1, \ldots, 10 \tag{9}$$

where $x_i$ in (9) is the input to the $i$th neuron in the input layer as indicated in Fig. 4.

For simplicity and to capture the local statistics, the 36 causal pixels in Fig. 5 are chosen empirically as the online training patterns. A larger training area can also be used for training to achieve better prediction, but with higher complexity. We found through experiments that a training area of 36 pixels gives a good trade-off between prediction and complexity. The online updating process is performed by iterative learning on the 36 training pixels. Most of the online updating processes converge within the first few cycles [32]. Therefore we set the learning rate $\eta$ as 0.9 and the momentum term $\alpha$ as 0 for the first three training cycles to avoid being trapped in the local minima, and 0.1 and 0.9, respectively after that to avoid the oscillation problem [32, 36]. The learning process will stop automatically whenever a mean square error tolerance of 0.00005 or a maximum training cycle of 20 is reached [36]. Updated weights are used for current prediction and passed on to the next coding pixel as initial weights.

## 4 Texture context matching

The ANP can achieve a very good prediction except in areas containing edges. However, for areas containing edges, convergence of the network weights is slow and the prediction error is large. For such areas, the encoder switches to TCM and estimates the coding pixel using pixels with similar contexts. To reduce the complexity of texture matching, we will use a shortened texture context.

### 4.1 Shortened texture context

The four pixels marked by $m_1, m_2, m_3, m_4$ as shown in Fig. 6 constitute the shortened texture context for TCM. They are the same pixels $x_1, x_2, x_3, x_4$ in Section 3 but are renamed as $m_1, \ldots, m_4$ for the convenience of explanation. As similar contexts are most likely to appear in the vicinity of the coding pixel, there is no need to perform an exhaustive search or near-global search in the causal area. The pixels to be used for context matching are the same 36 pixels used for the online training area of ANP for convenience (Fig. 6).

We first calculate the normalised shortened texture context

$$\left\{\frac{m_1}{l_m}, \frac{m_2}{l_m}, \frac{m_3}{l_m}, \frac{m_4}{l_m}\right\} \tag{10}$$

where $l_m$ is the length or root mean square (rms) value of the 4-dimensional shortened texture context, given as

$$l_m = \sqrt{\frac{1}{4}\sum_{i=1}^{4} m_i^2} \tag{11}$$

For each of the 36 pixels in the context matching area, we compute its normalised shortened texture context as in (10). We calculate the Euclidean distance $d_{mm'}$

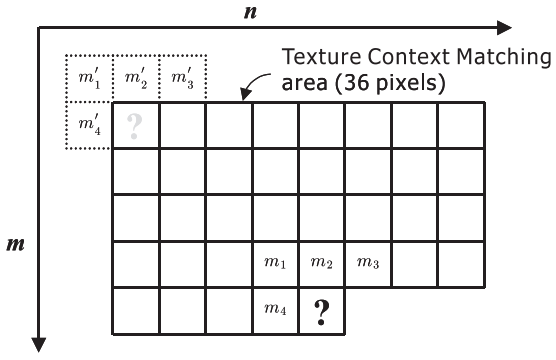


**Fig. 6** *Area for texture context matching*

between the normalised shortened texture context of the current pixel and that of each of the 36 pixels to be matched. The Euclidean distance $d_{mm'}$ between the two normalised shortened texture context is

$$d_{mm'} = \sum_{i=1}^{4} \left( \frac{m_i}{l_m} - \frac{m'_i}{l_{m'}} \right)^2 \tag{12}$$

We find the two pixels with the two shortest Euclidean distances. Suppose the values of these two pixels are $y_1$ and $y_2$, where $y_i$ has been normalised by the length of its shortened texture context. The coding pixel value $x$ is estimated as

$$x_p = l_m \left( \frac{y_1 + y_2}{2} \right) \tag{13}$$

For coding pixels on the boundaries of the image, the matching areas are smaller. Only one pixel in the matching area is chosen for estimating the coding pixel. With the length-normalisation process, contexts that have different amplitudes, but are otherwise similar, can be counted together after the length is normalised and the number of matches can be increased [33, 39].

## 5 Error compensation using automatic fuzzy context modelling

The prediction errors $x - x_p$ in the proposed SWAP system can be further refined by learning from previous predictions. Gathering prediction errors of similar contexts and calculating the sample means of errors in the same group, the current prediction can be further compensated. In the SWAP system, adaptive error modelling is achieved by designing codebooks using the proposed fuzzy context clustering. The context is dynamically generated and modified in the coding process. Also, the number of contexts is not fixed and will depend on the statistics of the image to be encoded.

Let $e_i$ be the uncompensated prediction error of $x_i$ in Fig. 3 for $i = 1, 2, 3, 4$. We define the compound context vector $\boldsymbol{v}^{(t)}$ of a coding pixel as

$$\boldsymbol{v}^{(t)} = \{x_1, x_2, \ldots, x_{10}, e_1, e_2, e_3, e_4\} \tag{14}$$

where $x_i$, $i = 1, \ldots, 10$ are as shown in Fig. 3 and $e_i$, $i = 1, \ldots, 4$ are the uncompensated prediction errors corresponding to $x_1, \ldots, x_4$, respectively. The pixels are encoded sequentially in raster scan order and the superscript $(t)$ in $\boldsymbol{v}^{(t)}$ denotes the pixel index. We have incorporated prediction errors in codebook designs, because the amount to be compensated is likely to be related to the prediction errors of neighbouring pixels. Furthermore, the compound context vector $\boldsymbol{v}^{(t)}$ of a coding pixel is likely to be related to existing clusters with some membership degrees. Therefore it is dynamically assigned to existing clusters or to a new cluster. In conventional fuzzy K-means clustering, the vectors to be classified are available all at once, and the number of clusters is fixed [36, 40]. Here, the compound context vectors appear sequentially. Therefore, the UFCL algorithm [37], which is suitable for sequential input vector classification, is used for the clustering process with some modifications. The number of clusters is now a variable. Furthermore, we will propose a learning rate that has the desired property of approaching zero when an optimal classification is achieved [37]. In this case, the cluster updating formula is in the form of a weighted summation. We call the modified UFCL algorithm 'fuzzy context clustering' in this paper.

### 5.1 Fuzzy context clustering

Assume we have $K$ clusters or codewords in the codebook currently. The $i$th codeword or $i$th cluster centre will be denoted by $c_i^{(t)}$. For an incoming compound context vector $\boldsymbol{v}^{(t+1)}$ of coding pixel, the distance $d_i^{(t+1)}$ between $\boldsymbol{v}^{(t+1)}$ and the existing $i$th cluster centre $c_i^{(t)}$ is calculated as

$$d_i^{(t+1)} = \left\| \boldsymbol{v}^{(t+1)} - c_i^{(t)} \right\|^2 \quad \text{for } i = 1, 2, \ldots, K \tag{15}$$

where $\|\cdot\|$ denotes the $L_2$-norm. If the minimum distance value $d_{\min} = \min_{i=1,\ldots,K} \{d_i^{(t+1)}\}$ is greater than a predefined threshold $\beta = 15\,000$, a new cluster is added. Therefore the number of codewords is increased to $K+1$, and the new cluster centre $c_{(K+1)}^{(t+1)} = \boldsymbol{v}^{(t+1)}$. If $d_{\min} < \beta$, $\boldsymbol{v}^{(t+1)}$ is used to update existing clusters based on the fuzzy context clustering algorithm. The membership degree $A_i^{(t+1)}$ of $\boldsymbol{v}^{(t+1)}$ to the $i$th cluster is defined by Looney [37] and Klir and Yuan [40]

$$A_i^{(t+1)} = \left\{ \sum_{j=1}^{K} \left[ \frac{d_i^{(t+1)}}{d_j^{(t+1)}} \right]^{1/(m-1)} \right\}^{-1}, \quad \text{for } i = 1, 2, \ldots, K \tag{16}$$

where the exponential weight $m$, which controls the degree of fuzziness, is chosen to be 1.25 [36, 40]. The membership degree $A_i^{(t+1)}$ is inversely proportional to the distance between $\boldsymbol{v}^{(t+1)}$ and the existing cluster centres. The fuzzy clustering algorithm converges to the classical $K$-means clustering when $m$ approaches one [36, 40]. We define $S_i^{(t)}$, the accumulated weight of the $i$th cluster, as

$$S_i^{(t)} = \sum_{n=1}^{t} (A_i^n)^m, \quad \text{for } i = 1, 2, \ldots, K \tag{17}$$

Note that $S_i^{(t)}$ is very similar to the membership degree summation of vectors with respect to the same cluster in a conventional fuzzy partition matrix. Existing cluster centres are updated by Looney [37]

$$c_i^{(t+1)} = c_i^{(t)} + \lambda^{(t+1)} (A_i^{(t+1)})^m (\boldsymbol{v}^{(t+1)} - c_i^{(t)}), \quad \text{for } i = 1, 2, \ldots, K \tag{18}$$

where $\lambda^{(t+1)}$ is the learning rate, initialised to one. We will choose

$$\lambda^{(t+1)} = \frac{1}{S_i^{(t)} + (A_i^{(t+1)})^m} \tag{19}$$

The resulting learning rate $\lambda^{(t+1)}$ will become smaller as coding proceeds and approaches zero when an optimal classification is reached. Using (19), (18) can be rewritten as

$$c_i^{(t+1)} = \frac{S_i^{(t)} \cdot c_i^{(t)} + (A_i^{(t+1)})^m \cdot \boldsymbol{v}^{(t+1)}}{S_i^{(t)} + (A_i^{(t+1)})^m}, \quad \text{for } i = 1, 2, \ldots, K \tag{20}$$

### 5.2 Error estimate

The value $e_p$ to be used in compensating the prediction error of the coding pixel is given by

$$e_p = \sum_{i=1}^{K} A_i^{(t+1)} \cdot e_i^{(t)} \tag{21}$$

where $e_i^{(t)}$ is the sample mean of prediction errors in the $i$th cluster. We update $e_i^{(t)}$ [37] using

$$e_i^{(t+1)} = e_i^{(t)} + \lambda^{(t+1)}(A_i^{(t+1)})^m(x^{(t+1)} - x_p^{(t+1)} - e_i^{(t)}),$$
$$\text{for } i = 1, 2, \ldots, K \tag{22}$$

Similarly, using (19), (22) can be written as

$$e_i^{(t+1)} = \frac{S_i^{(t)} \cdot e_i^{(t)} + (A_i^{(t+1)})^m \cdot (x^{(t+1)} - x_p^{(t+1)})}{S_i^{(t)} + (A_i^{(t+1)})^m},$$
$$\text{for } i = 1, 2, \ldots, K \tag{23}$$

We now form a more refined prediction $x_{cpd} = x_p + e_p$, where $x_p$ is the output of predictor ANP or TCM.

## 6 Experiments

Experimental results of the proposed SWAP coder and comparisons to existing state-of-the-art linear and non-linear predictors are given in this section. All the test images used in the experiments are the same as those in Li and Orchard [22], Ye *et al.* [26], Meyer and Tischer [28] and Manikopoulos [30]. They are widely used as the test images in lossless compression of still images. Moreover, they are obtained from the TMW website http://www.csse.monash.edu.au/~bmeyer/tmw/ [28]. The parameters used in the predictor switch, ANP and TCM, are the same in all the experiments. We have found the set of parameters working well for all types of images. We first demonstrate the usefulness of the individual blocks in the SWAP system and then present the bit rate performance of the SWAP system. The SWAP system has four primary components. As the ANP is well-known, we will only address the remaining three: the predictor switch, the TCM predictor and the error modelling.

### 6.1 Predictor switch

To demonstrate the effectiveness of the 'predictor switch', we use the image 'Shapes' (Fig. 7), which is an artificial image with many edges and lines obtained from Meyer and Tischer [28]. The pixels that satisfy the two conditions in (2) are marked in Fig. 8. We can see from Fig. 8 that the predictor switch box has successfully picked out the pixels around edges. As another example, we apply the predictor switch to the image 'Noisesquare' (Fig. 9), an image with salt-and-pepper noise. The pixels for which the TCM are used are as shown in Fig. 10. We see from Fig. 10 that the predictor switch is robust to moderate salt-and-pepper noise.

In addition to artificial images, we also apply the predictor switch to the natural images, 'Airplane' (Fig. 11*a*) and 'Lennagrey' (Fig. 11*b*). As can be seen in Figs. 12*a* and 12*b*, the pixels around edges in the images 'Airplane' and 'Lennagrey' have been picked out successfully.

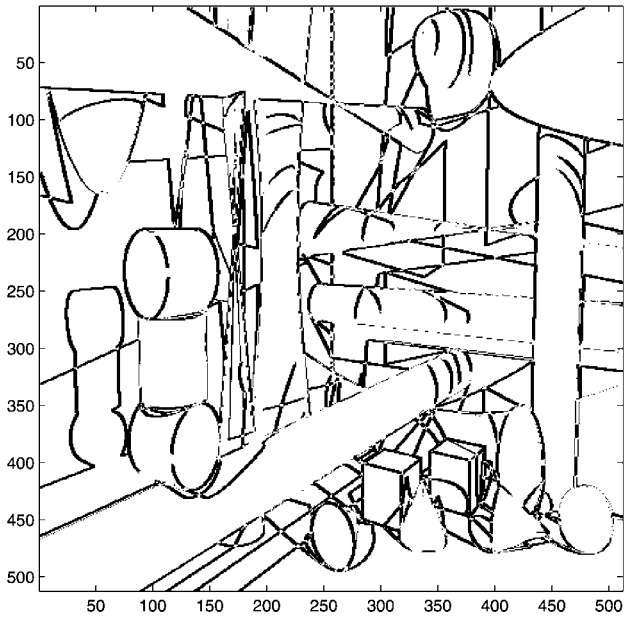


**Fig. 8** *Pixels for which TCM is used in the image 'Shapes'*

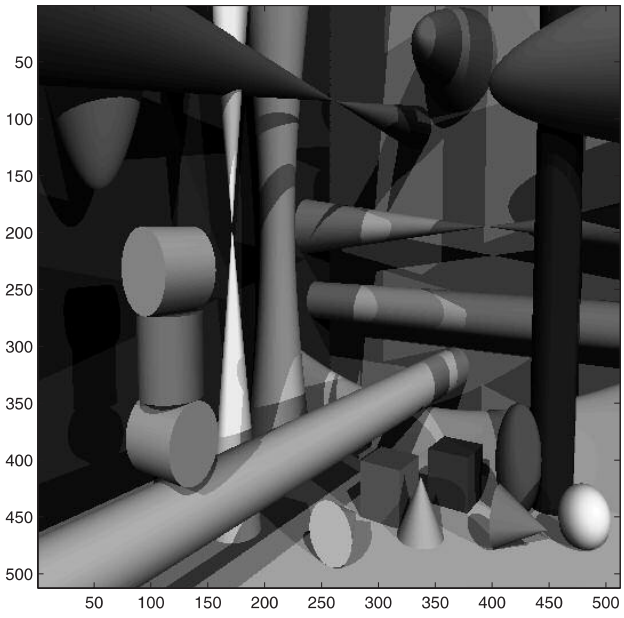


**Fig. 7** *Image 'Shapes'*

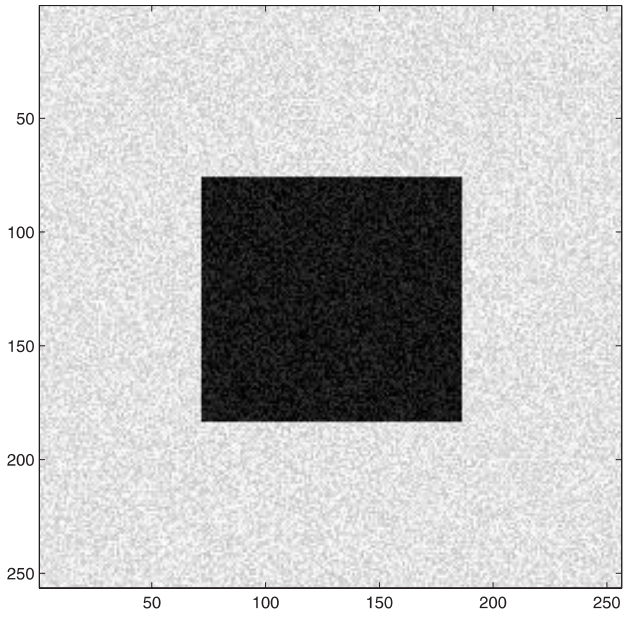


**Fig. 9** *Image 'Noisesquare'*

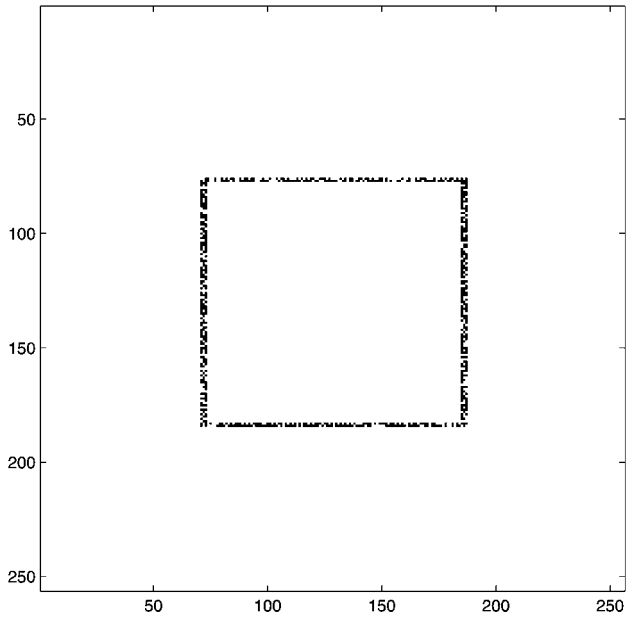


**Fig. 10** *Pixels for which TCM is used in the image 'Noisesquare'*

### 6.2 Error modelling

Error compensation relies on effective error modelling. The effect of error modelling can be demonstrated by observing the histograms of compensated and uncompensated error. For the image 'Lennagrey', we show in Figs. 13 and 14, respectively, the compensated prediction error and the corresponding histogram. The statistical redundancy is removed efficiently as can be seen in Figs. 13 and 14. The usefulness of the proposed automatic fuzzy context modelling for error refinement can be best observed in Fig. 14, in which the histogram of prediction errors with and without error compensation are shown. In Fig. 14, the first-order entropy for compensated errors is 3.97 bits and 4.20 bits for uncompensated errors.

### 6.3 TCM predictor

The usefulness of the proposed TCM predictor can be demonstrated by the following experiment. We use the image 'Shapes' in Fig. 7. The pixels for which TCM is used as the predictor are shown in Fig. 8. The histogram of uncompensated prediction errors for those pixels using TCM is shown in Fig. 15. For comparison, we also show in Fig. 15 the histogram of uncompensated prediction error if ANP were used for those pixels instead. The histogram with TCM is much narrower than that with ANP; TCM has a smaller prediction error than ANP does around edges. Indeed, the entropies corresponding to the two histograms in Fig. 15 are, respectively, 2.97 bits (TCM) and 5.33 bits (ANP).

### 6.4 Performance of the SWAP system

Table 1 gives comparisons of uncompensated prediction errors for a set of eight test images in first-order entropies. The results of an MED [17, 18], a GAP [5, 16] and an edge directed predictor (EDP) with different orders are taken from Li and Orchard [22]. As can be seen in Table 1, the proposed SWAP system can remove the statistical redundancy efficiently. It achieves a noticeable improvement when compared with MED and GAP. The SWAP gives lower entropies in five out of the eight test images when compared with those of EDP [22].

Table 2 gives the actual bit rates by CALIC [16] and TMW [28] for a set of fourteen test images obtained from the website of TMW [28]. Results listed in the last two columns of Table 2 are taken directly from Wu and Memon [16] and Meyer and Tischer [28]. The compensated prediction errors are coded using a conditional arithmetic coder adapted from Witten *et al.* [38] in the SWAP system. All the bit rates reported by SWAP are obtained using the same parameters described in previous sections and no individual optimisation is performed. Table 2 shows that SWAP achieves lower bit rates than CALIC [16] in ten out of the fourteen test images. Encouragingly,

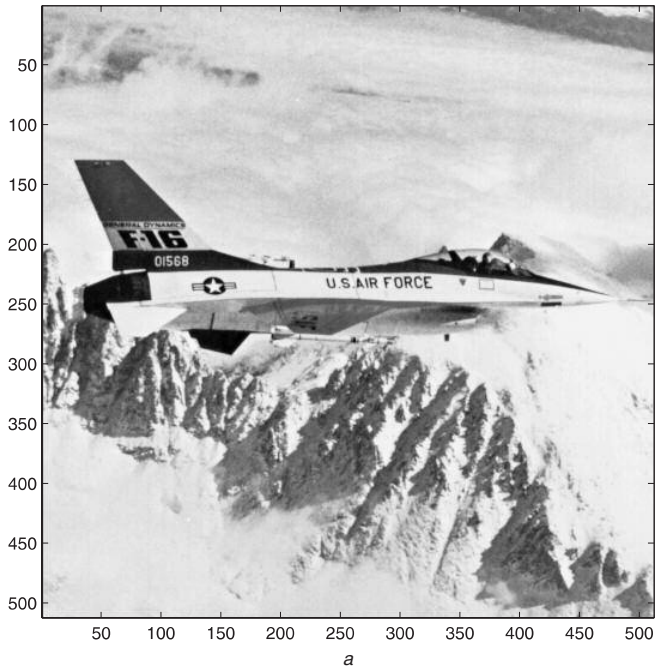



**Fig. 11** *Natural images*
*a* Airplane
*b* Lennagrey

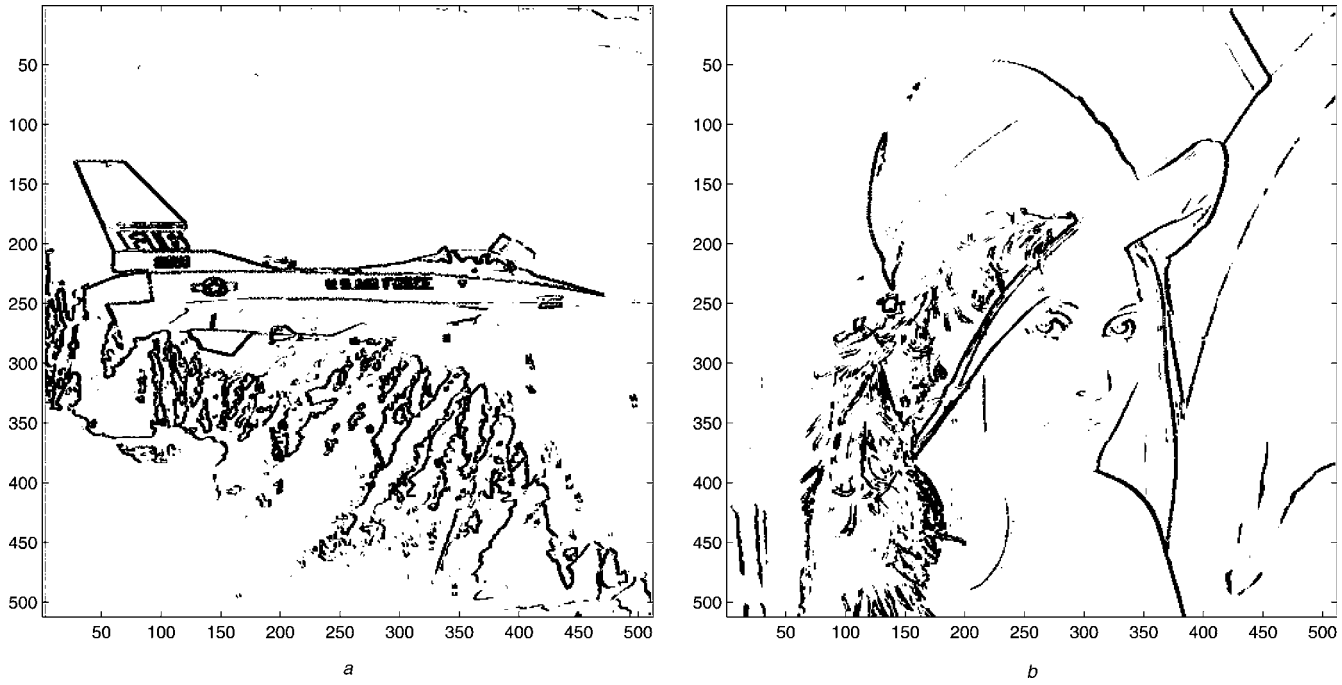


**Fig. 12** *Pixels for which TCM is used*

*a* Airplane
*b* Lennagrey

SWAP achieves bit rates lower than the highly complex TMW in three images 'Airplane', 'Balloon' and 'Noisesquare'.

### 6.5 Side information of the proposed SWAP coder

The only side information to be transmitted is the dimension, that is the height and the width of the image. Furthermore, the SWAP coder uses only causal pixels for estimating the coding pixels; no additional side information needs to be transmitted. It is noted that the proposed SWAP coder is symmetric, meaning the decoder has the same predictor switch as the encoder, and perform ANP/TCM prediction and error compensation just like the encoder. As indicated earlier, all the results presented in the SWAP coder are obtained using the same parameters, no optimisation is performed with respect to individual images. The parameters that are used in the encoder are also available in the decoding process, no additional information needs to be transmitted.

### 6.6 Trade-off between computational complexity and performance

For complexity comparison, we have listed in Table 3 the operation counts of state-of-the-art MED and GAP predictor. As can be seen in Table 3, the MED, a division and multiplication-free predictor that has been standardised into JPEG-LS [17, 18], has the lowest complexity. Though conceptually simple, the performance of the MED predictor is surprisingly good. The GAP, employed by the CALIC

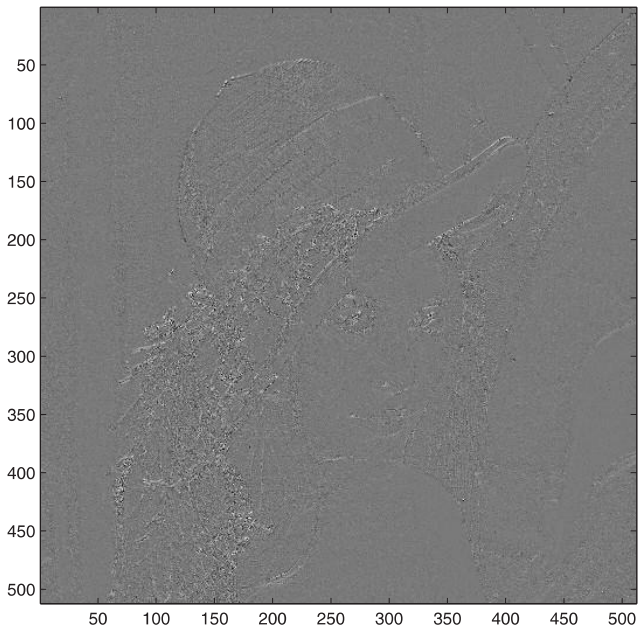


**Fig. 13** *Image of compensated prediction errors for 'Lennagrey'*

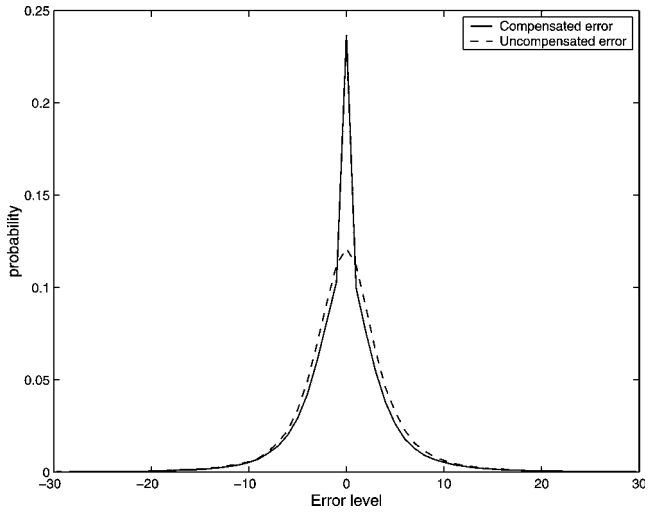


**Fig. 14** *Histogram of prediction errors for 'Lennagrey'*

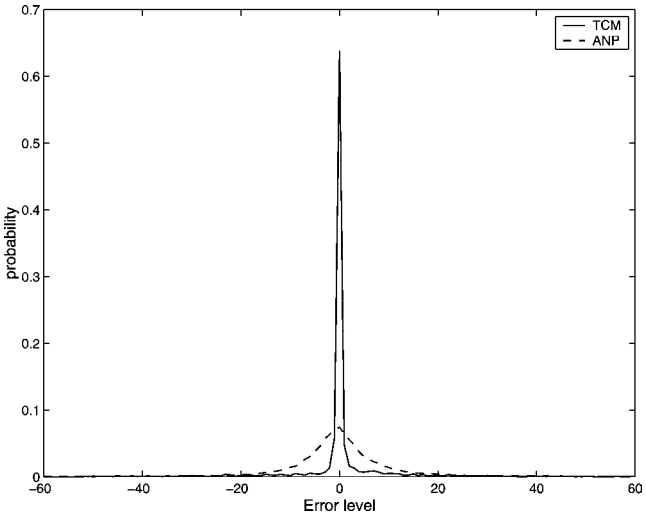


**Fig. 15** *Histogram of prediction errors for the pixels shown in Fig. 8*

coding system [16], is an adaptive, non-linear predictor. The GAP calculates the gradient of a coding pixel and uses one out of a set of seven predictors based on the calculated gradient. As can be seen in Table 3, the GAP has higher complexity than that of MED predictor. Both the coefficients of MED and GAPs are fixed during the coding process.

The EDP [22] adapts itself on-the-fly using LS optimisation to update predictor coefficients and has a higher complexity than MED and GAP predictors. It is noted that the normal equations provide the key for the solution of an LS problem [41, 42]. Numerically, the normal equations can be solved by Cholesky decomposition or singular value decomposition (SVD) depending on the singularity of the normal equations [41, 42]. For the normal equations to be non-singular, the Cholesky decomposition, which requires about half the usual number of multiplications than alternative methods, can be used; otherwise the SVD is applied. For this, Kau and Lin [23] have pointed out that most of the LS adaptation processes can be solved by using Cholesky decomposition and it requires $N^3/6$ multiplications, where $N$ is the prediction order [41, 42]. Therefore the major complexity of the LS adaptation process in EDP will be in constructing the normal equations rather than solving them [41, 42]. In addition to EDP, the TMW [28], a two-pass coder that uses multiple linear predictors and global image analysis, has achieved the lowest bit rates in most images when compared to state-of-the-art linear and non-linear predictors. However, the use of a highly complex image analysis process as well as the blending of probability distributions leads to the high computational cost in TMW. The complexity of TMW is much higher than that of EDP (seconds against hours) [22].

In the SWAP system, ANP and the error models are updated in the coding process. The proposed SWAP system needs higher complexity than EDP but lower complexity than TMW. The performance of the SWAP system is also in between EDP and TMW. There are four major blocks, 'predictor switch', 'ANP', 'TCM predictor' and 'fuzzy context modelling', in the proposed SWAP system. To analyse the computational complexity, the operation counts for each coding pixel in the four blocks are listed in Table 4. The predictor switch, performing boundary detection operation, determines which of the predictors, ANP or TCM, is to be used. If ANP is chosen, the online training of ANP is performed ((5) to (9)). The operation counts of one training pixel for online training of ANP are listed in Table 4.

As can be seen in Table 4, the major computational cost in the proposed SWAP would be in the online training of ANP. Indeed, the number of multiplications required for a back propagation neural network is proportional to the number of connection weights, that is, the time complexity is O($N$), where $N$ is the number of connection weights [43]. In this paper, the number of neurons in the input layer and the hidden layer is ten and five, respectively. Therefore it requires 237, that is, $h(4i + 7) + 2$, multiplications during the online training of ANP for one training pattern. The actual multiplications required should be multiplied by the number of training patterns and training cycles. To reduce the time complexity during the online training of ANP, the training pattern is empirically chosen to be the

**Table 1: First-order entropies of prediction errors**

| Image | Proposed SWAP | MED [17] | GAP [16] | EDP [22] $N = 6$ | EDP [22] $N = 10$ |
|---|---|---|---|---|---|
| Baboon | 6.02 | 6.28 | 6.22 | 6.01 | 5.99 |
| Lena | 4.54 | 4.90 | 4.75 | 4.60 | 4.58 |
| Lennagrey | 4.20 | 4.56 | 4.40 | 4.26 | 4.22 |
| Peppers | 4.45 | 4.95 | 4.78 | 4.52 | 4.50 |
| Barb | 4.36 | 5.21 | 5.15 | 4.44 | 4.35 |
| Barb2 | 4.81 | 5.19 | 5.06 | 4.80 | 4.78 |
| Boats | 4.03 | 4.31 | 4.29 | 4.14 | 4.10 |
| Goldhill | 4.57 | 4.72 | 4.70 | 4.60 | 4.58 |
| Average | 4.62 | 5.02 | 4.92 | 4.67 | 4.64 |

**Table 2: Comparisons with existing lossless image coders (in bits/sample)**

| Image | First order entropy | Cluster number | SWAP bit rate | CALIC bit rate [16] | TMW bit rate [28] |
|---|---|---|---|---|---|
| Airplane | 6.7058 | 675 | 3.58 | 3.74 | 3.60 |
| Baboon | 7.3579 | 1392 | 5.86 | 5.88 | 5.73 |
| Balloon | 7.3258 | 351 | 2.49 | 2.83 | 2.66 |
| Barb | 7.4664 | 712 | 4.12 | 4.32 | 4.09 |
| Barb2 | 7.4838 | 842 | 4.55 | 4.53 | 4.38 |
| Boats | 7.095 | 591 | 3.64 | 3.83 | 3.61 |
| Camera | 7.009 | 1958 | 4.39 | 4.19 | 4.10 |
| Couple | 6.3902 | 453 | 3.75 | 3.61 | 3.45 |
| Goldhill | 7.6232 | 429 | 4.30 | 4.39 | 4.27 |
| Lena | 7.594 | 676 | 4.35 | 4.48 | 4.30 |
| Lennagrey | 7.4473 | 476 | 3.95 | 4.11 | 3.91 |
| Noisesquare | 5.723 | 309 | 5.16 | 5.44 | 5.54 |
| Peppers | 7.5924 | 734 | 4.25 | 4.42 | 4.25 |
| Shapes | 6.7395 | 877 | 1.56 | 1.14 | 0.76 |

**Table 3: Operation counts for MED and GAP predictors**

| Operation | Comparison | ADD/SUB | MUL/DIV | ABS |
|---|---|---|---|---|
| MED | ≤3 | ≤2 | | |
| GAP | ≤6 | ≤18 | ≤7 | 6 |

**Table 4: Counts of different operations for 'boundary detection', 'ANP', 'TCM prediction' and 'fuzzy context modelling' in proposed SWAP system**

| Operation | Comparison | ADD/SUB | MUL/DIV | Square | Square root | Taking inverse | Taking exponent |
|---|---|---|---|---|---|---|---|
| Edge detection | $n+2$ | $\leq 4n$ | $\leq 7$ | $\leq (n+3)$ | | | |
| ANP | | $h(i+1)$ | $h(i+1)$ | | | $h+1$ | $h+1$ |
| Online training of ANP for one training pattern | | $h(3i+4)+2$ | $h(4i+7)+2$ | | | $h+1$ | $h+1$ |
| TCM prediction | $(a-1)(a-2)$ | $at-a+3t-1$ | $at+a+t+3$ | $at+2t$ | $a+1$ | | |
| Error estimate | | $k(2c+1)-2$ | $2k$ | $c$ | 1 | $k$ | $k$ |
| Context model update | | $k(3c+2)$ | $4kc$ | | | | $3k$ |

$n$ is the number of pixels in texture context
$i$ is the number of neurons in the input layer
$h$ is the number of neurons in the hidden layer
$t$ is the number of pixels in the shortened texture context
$a$ is the number of pixels in the texture context matching area
$k$ is the number of contexts
$c$ is the dimension of the compound context vector

36 pixels in Fig. 5, and the maximum training cycle is confined to 20. Fortunately, most of the online training processes converge within the first few training cycles [32]. It is noted that the execution time of the proposed SWAP coder can be accelerated with a little degradation in the actual bit rate by setting an error threshold such that, the network will not perform online training until the prediction error exceeds the predefined threshold. Moreover, increasing the number of iterations and the training areas gives marginal improvement in entropies at the cost of a drastic increase in the training time period. Therefore, the maximum number of iterations and the number of training patterns, as well as the number of neurons in the hidden layer are determined experimentally and represent a compromise between the performance and the training time.

Though the proposed switching predictor structure has higher complexity than that of linear predictors, it has been shown to provide better performance in terms of prediction results when compared with linear predictors. Indeed, many approaches have been proposed so that the computational burden can be reduced by using the back propagation neural network [44, 45]. A low complexity fuzzy activation function composed of a linear function and constants has been proposed by Soria-Olivas *et al.* [44] so that the non-linear operation in calculating the sigmoid function can be avoided. Besides, we know that the multiplication between two integers can be substituted by a shift, provided one of them is a power of two. Therefore a learning procedure based on the power-of-two approach has been proposed by Marchesi *et al.* [45] with only shift and add operations. The multiplication-free algorithm can be applied in both the forward and learning phases of the network, and can be easily implemented with digital hardware [45]. By the non-linear nature, multi-layer neural networks are well suited to the task of image compression, and the run time performance is expected to be improved with the rapid advances in computational facilities and the development of very large scale integration (VLSI) implementations [29].

## 7 Conclusion

In this paper, a new coder called SWAP is proposed. The SWAP system switches between two predictors, ANP and TCM automatically. To decide which predictor is to be used, we also propose a switching criterion that can pick out pixels around edges effectively. The ANP making non-linear prediction using a neural network, performs very well in slowly varying areas. The TCM provides a very nice complement to ANP. As the simulation example has demonstrated, the TCM predictor can achieve very good prediction around edges, where ANP tends to have larger prediction errors. For error refinement, automatic context modelling is achieved using fuzzy context clustering which leads to a modelling of errors that adapts itself to the input statistics. The usefulness of the proposed SWAP system is demonstrated through the reduction of first-order entropy and actual bit rate in the tested images.

## 8 Acknowledgment

This work was supported by NSC 93-2213-E-009-115.

## 9 References

1 Sayood, K.: 'Introduction to data compression' (Morgan-Kaufmann, 2000, 2nd edn.), pp. 139–163
2 Rao, K.R., and Hwang, J.J.: 'Techniques and standards for image, video and audio coding' (Prentice-Hall, 1996), pp. 31–42
3 Carpentieri, B., Weinberger, M.J., and Seroussi, G.: 'Lossless compression of continuous-tone images', *Proc. IEEE*, 2000, **88**, (11), pp. 1797–1809
4 Memon, N., and Wu, X.: 'Recent developments in context-based predictive techniques for lossless image compression', *Comput. J.*, 1997, **40**, (2/3), pp. 127–136
5 Wu, X.: 'An algorithmic study on lossless image compression'. Proc. Data Compression Conf., Snowbird, UT, USA, 31 March–3 April 1996, pp. 150–159
6 Wu, X., and Qiu, T.: 'Wavelet coding of volumetric medical images for high throughput and operability', *IEEE Trans. Med. Imaging*, 2005, **24**, (6), pp. 719–727
7 Deever, A.T., and Hemami, S.S.: 'Lossless image compression with projection-based and adaptive reversible integer wavelet transforms', *IEEE Trans. Image Process.*, 2003, **12**, (5), pp. 489–499
8 Grangetto, M., Magli, E., Martina, M., and Olmo, G.: 'Optimization and implementation of the integer wavelet transform for image coding', *IEEE Trans. Image Process.*, 2002, **11**, (6), pp. 596–604
9 Boulgouris, N.V., Tzovaras, D., and Strintzis, M.G.: 'Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding', *IEEE Trans. Image Process.*, 2001, **10**, (1), pp. 1–14
10 Reichel, J., Menegaz, G., Nadenau, M.J., and Kunt, M.: 'Integer wavelet transform for embedded lossy to lossless image compression', *IEEE Trans. Image Process.*, 2001, **10**, (3), pp. 383–392
11 Avcibas, I., Memon, N., Sankur, B., and Sayood, K.: 'A successively refinable lossless image-coding algorithm', *IEEE Trans. Commun.*, 2005, **53**, (3), pp. 445–452

12 Memon, N., Kong, X., and Cinkler, J.: 'Context-based lossless and near-lossless compression of EEG signals', *IEEE Trans. Inf. Technol. Biomed.*, 1999, **3**, pp. 231–238
13 Yang, K.H., and Faryar, A.F.: 'A context-based predictive coder for lossless and near-lossless compression of video'. Proc. IEEE Int. Conf. Image Process., 10–13 September 2000, vol. 1, pp. 144–147
14 Das, M.K., Li, C.C., and Burgett, S.R.: 'A new multiresolution predictive scheme for lossless compression of medical images'. Proc. IEEE Int. Symp. Computer-Based Medical Systems, 13–16 June 1993, pp. 34–39
15 Lin, X.Y., Shark, L.-K., Varley, M.R., and Matuszewski, B.J.: 'A hybrid lossless compression scheme using region-based predictive coding and integer wavelet transform'. Proc. IEEE Int. Conf. Image Process., 7–10 October 2001, vol. 3, pp. 486–489
16 Wu, X., and Memon, N.: 'Context-based, adaptive, lossless image coding', *IEEE Trans. Commun.*, 1997, **45**, (4), pp. 437–444
17 Weinberger, M.J., Seroussi, G., and Sapiro, G.: 'The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS', *IEEE Trans. Image Process.*, 2000, **9**, (8), pp. 1309–1324
18 ISO/IEC JTC 1/SC 29/WG 1 FCD 14495 – public draft: 'FCD 14495, lossless and near-lossless coding of continuous tone still images (JPEG-LS)', July 1997
19 Motta, G., Storer, J.A., and Carpentieri, B.: 'Lossless image coding via adaptive linear prediction and classification', *Proc. IEEE*, 2000, **88**, (11), pp. 1790–1796
20 Yu, T.-H.: 'A fuzzy logic-based predictor for predictive coding of images', *IEEE Trans. Fuzzy Syst.*, 1998, **6**, (1), pp. 153–162
21 Kuroki, N., Nomura, T., Tomita, M., and Hirano, K: 'Lossless image compression by two-dimensional linear prediction with variable coefficients', *IEICE Trans. Fundam.*, 1992, **E75-A**, (7), pp. 882–889
22 Li, X., and Orchard, M.T.: 'Edge-directed prediction for lossless compression of natural images', *IEEE Trans. Image Process.*, 2001, **10**, (6), pp. 813–817
23 Kau, L.-J., and Lin, Y.-P.: 'Adaptive lossless image coding using least squares optimization with edge-look-ahead', *IEEE Trans. Circuits Syst. II*, 2005, **52**, (11), pp. 751–755
24 Ye, H., Deng, G., and Devlin, J.C.: 'A weighted least squares method for adaptive prediction in lossless image compression'. Proc. Picture Coding Symp., Saint-Malo, France, 2003, pp. 489–493
25 Meyer, B., and Tischer, P.E.: 'Glicbawls – grey level image compression by adaptive weighted least squares'. Proc. Data Compression Conf., Snowbird, UT, USA, 2001, pp. 503–514
26 Ye, H., Deng, G., and Devlin, J.C.: 'Adaptive linear prediction for lossless coding of greyscale images'. Proc. IEEE Image Process. Conf., 2000, Vol. 1, pp. 128–131
27 Wu, X.: 'Lossless compression of continuous-tone images via context selection, quantization, and modeling', *IEEE Trans. Image Process.*, 1997, **6**, (5), pp. 656–664
28 Meyer, B., and Tischer, P.E.: 'TMW—A new method for lossless image compression'. Proc. Int. Picture Coding Symp., Berlin, Germany, October 1997
29 Dony, R.D., and Haykin, S.: 'Neural network approaches to image compression', *Proc. IEEE*, 1995, **83**, (2), pp. 288–303
30 Manikopoulos, C.N.: 'Neural network approach to DPCM system design for image coding', *IEE Proc. I, Commun. Speech Vis.*, 1992, **139**, (5), pp. 501–507
31 Li, H.: 'Some new results in nonlinear predictive image coding using neural networks'. Proc. Neural Networks for Signal Processing Workshop, Helsingoer, Denmark, August 1992, pp. 411–420
32 Marusic, S., and Deng, G.: 'A neural network based adaptive non-linear lossless predictive coding technique'. Proc. Fifth Int. Symp. on Signal Processing and Its Applications, 1999, vol. 2, pp. 653–656
33 Slyz, M.J., and Neuhoff, D.L.: 'A nonlinear VQ-based predictive lossless image coder'. Proc. Data Compression Conf., March 1994, pp. 304–310
34 Hornik, K., Stinchcommbe, M., and White, H.: 'Multilayer feedforward networks are universal approximators', *Neural Netw.*, 1989, **2**, pp. 359–366
35 Sontag, E.D.: 'Feedback stabilization using two-hidden-layer nets', *IEEE Trans. Neural Netw.*, 1992, **3**, (6), pp. 981–990
36 Lin, C.-T., and Lee, C.S.G.: 'A neural-fuzzy synergism to intelligent systems' (Prentice-Hall, 1996), pp. 235–250
37 Looney, C.G.: 'Pattern recognition using neural networks' (Oxford, 1997), pp. 250–256
38 Witten, I.H., Neal, R.M., and Cleary, J.G.: 'Arithmetic coding for data compression', *Commun. ACM*, 1987, **30**, (6), pp. 520–540
39 Gersho, A., and Gray, R.: 'Vector quantization and signal compression' (Kluwer Academic Publishers, Boston, 1992)
40 Klir, G.J., and Yuan, B.: 'Fuzzy sets and fuzzy logic, theory and applications' (Prentice-Hall, Int. Editions, 1995), pp. 358–365
41 Leon, S.J.: 'Linear algebra with applications' (Prentice-Hall, New Jersey, 2002), pp. 482–488
42 Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P.: 'Numerical recipes in C' (Cambridge University Press, 2002), pp. 96–98. Available online at: http://www.library.cornell.edu/nr/cbookcpdf.html.
43 Yousefi'zadeh, H., and Jonckheere, E.A.: 'Dynamic neural-based buffer management for queuing systems with self-similar characteristics', *IEEE Trans. Neural Netw.*, 2005, **16**, (5), pp. 1163–1173
44 Soria-Olivas, E., Martin-Guerrero, J.D., Camps-Valls, G., Serrano-Lopez, A.J., Calpe-Maravilla, J., and Gomez-Chova, L.: 'A low-complexity fuzzy activation function for artificial neural networks', *IEEE Trans. Neural Netw.*, 2003, **14**, (6), pp. 1576–1579
45 Marchesi, M.L., Piazza, F., and Uncini, A.: 'Backpropagation without multiplier for multilayer neural networks', *IEE Proc., Circuits Devices Syst.*, 1996, **143**, (4), pp. 229–232