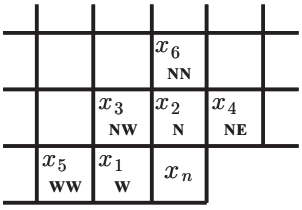

Fig. 2. Texture context around the coding pixel.

where t denotes the discrete coding sequence. Obviously, z_1 measures the strength of vertical deviation (i.e., horizontal edge detection), while z_2 is for that of horizontal deviation (i.e., vertical edge detection).

Layer1: Input layer.

The nodes in this layer just perform the buffering operations. That is, the input variables z_1 and z_2 are transmitted to layer2 directly.

Layer2: Membership layer.

Each node in this layer performs the membership degree measurement of a Gaussian function, and its output, μ_{ij}, specifies the degree to which the given input z_i satisfies the fuzzy quantifier, that is,

$$\mu_{ij} = exp\{-\frac{(z_i - m_{ij})^2}{\sigma_{ij}^2}\} \text{ for } i = 1, 2, \text{ and } j = 1, 2, 3, \quad (2)$$

where m_{ij} and σ_{ij} denotes the center and the width (i.e., standard deviation) of the Gaussian membership function respectively, and the subscript ij indicates the jth node associated with the ith input z_i in layer1.

Layer3: Rule layer.

The links in this layer are used to implement the antecedent matching. The matching operation (or the fuzzy "**AND**" aggregation) is chosen as the simple "**product**" operation instead of "**min**" operation. Therefore, each node in this layer multiplies incoming signals and sends the product to its output. The output of the kth node in this layer represents the firing strength of the kth rule, and is given by

$$\alpha_k = O_k^{(3)} = \mu_{1p} * \mu_{2q} = net_k^{(3)} \quad \text{for } k = 1, 2, \ldots, 9, \\ p = 1, 2, 3, \text{ and } 1 \le q = k - 3(p-1) \le 3. \quad (3)$$

Layer4: Output layer.

This layer performs the defuzzification process to get the numerical output. As can be seen in Fig. 1, the output of each node in layer3 (i.e., the firing strength α_k) is first weighted by a factor w_k, and then summed up together as the net input of layer4. The connection weight w_k, which represents the output action for the kth rule, is given by

$$w_k = A_k x(t) + B_k u(t) \quad \text{for } k = 1, 2, \ldots, 9, \quad (4)$$

where $x(t) = [x_1, x_2, \ldots, x_6]^T$ is composed of the six causal neighbors around the coding pixel, $A_k = [a_k^1, a_k^2, \ldots, a_k^6]^T$ can be regarded as the sixth-order predictor coefficients associated with the kth rule, $u(t) = [u_1, u_2]^T = [x_1 - x_3, x_2 - x_3]^T = z(t)$ is used for horizontal and vertical edge detection, and the vector $B_k = [b_1, b_2]^T$ is the weighting coefficients associated with the kth rule for the vector $u(t)$. As we know that a large prediction error can take place around an edge during the coding process, and this can be regarded as a step command in control system. Therefore, the term $B_k u(t)$, which is called "**P-controller**" in control system, is applied in (4) for prediction error suppression. It should be noted that the proposed P-controller compensator is quite different to the so-called bias cancelation technique in [1], [2], and both of which can be applied jointly for reduction of prediction errors.

The output of the predictor network is then a linear combination of the individual output of the rules in layer3 (Fig. 1), and is given by

$$y(t) = net^{(4)} = \sum_{k=1}^{9} \alpha_k w_k = \sum_{k=1}^{9} \alpha_k (A_k x + B_k u). \quad (5)$$

III. Network adaptation process

To adapt the parameters of the proposed predictor network during the coding process, we define a cost function $E(t)$ as

$$E(t) = \frac{1}{2}(d(t) - y(t))^2, \quad (6)$$

where $d(t)$ denotes the desired output and $y(t)$ denotes the actual output of the proposed TS-FNN system, and the back propagation (BP) learning algorithm is used to minimize the cost function. We will first derive the adaptation rules for the parameters in the consequent part. After that, adaptation rules for those parameters in layer2 will be given.

A-1. Consequent part: Derivation for Δa_k^r.

In the consequent part, coefficients of the nine sixth-order predictors and the P-controller compensators are to be updated. The adaptation rule for a_k^r, the rth coefficient of the sixth-order predictor associated with the kth rule (or kth predictor) is given by

$$a_k^r(t+1) = a_k^r(t) + \Delta a_k^r(t) \quad \text{for } r = 1, 2, \ldots, 6, \\ \text{and } k = 1, 2, \ldots, 9, \quad (7)$$

where $\Delta a_k^r(t)$ is the correction term for $a_k^r(t)$ given by

$$\begin{aligned} \Delta a_k^r(t) &= -\eta_A \frac{\partial E}{\partial a_k^r} \\ &= -\eta_A \left[\frac{\partial E}{\partial y}\right] \left[\frac{\partial y}{\partial net^{(4)}}\right] \left[\frac{\partial net^{(4)}}{\partial a_k^r}\right] \\ &= \eta_A \delta^{(4)} \alpha_k x_r, \end{aligned} \quad (8)$$

where η_A is the learning rate for a_k^r, and $\delta^{(4)}$ is the *error signal* of layer4 defined by

$$\delta^{(4)} = -\frac{\partial E}{\partial net^{(4)}} = -\left[\frac{\partial E}{\partial O^{(4)}}\right] \left[\frac{\partial O^{(4)}}{\partial net^{(4)}}\right] = (d - y). \quad (9)$$

A-2. Consequent part: Derivation for Δb_k^s**.**

With a similar approach for that of a_k^r, the correction term for b_k^s, i.e., the sth coefficient of the compensator associated with the kth rule, is given by

$$\Delta b_k^s(t) = -\eta_B \frac{\partial E}{\partial b_k^s} = \eta_B \delta^{(4)} \alpha_k u_s \quad \text{for } s = 1, 2, \text{ and } k = 1, 2, \ldots, 9, \tag{10}$$

where η_B is the learning rate for b_k^s.

B-1. Primary part: Derivation for Δm_{ij}**.**

In the primary part, the mean m_{ij} and the standard deviation σ_{ij} of the six Gaussian membership functions in layer2 have to be updated. The correction term for m_{ij}, i.e., the mean of the jth Gaussian membership function associated with the ith input variable, is given by

$$\begin{aligned} \Delta m_{ij}(t) &= -\eta_m \frac{\partial E}{\partial m_{ij}} \\ &= -\eta_m \left[\frac{\partial E}{\partial y}\right]\left[\frac{\partial y}{\partial net^{(4)}}\right]\left[\frac{\partial net^{(4)}}{\partial m_{ij}}\right] \\ &= \eta_m (d - y)\left[\frac{\partial net^{(4)}}{\partial m_{ij}}\right], \end{aligned} \tag{11}$$

where where η_m is the learning rate, and $\frac{\partial net^{(4)}}{\partial m_{ij}}$, by applying chain rule, can be expressed as

$$\begin{aligned} \frac{\partial net^{(4)}}{\partial m_{ij}} &= \sum_{k=1}^{9} \Phi_k^{(ij)} \left[\frac{\partial net^{(4)}}{\partial O_k^{(3)}}\right]\left[\frac{\partial O_k^{(3)}}{\partial net_k^{(3)}}\right]\left[\frac{\partial net_k^{(3)}}{\partial O_{ij}^{(2)}}\right]\left[\frac{\partial O_{ij}^{(2)}}{\partial m_{ij}}\right] \\ &= \frac{2(z_i - m_{ij})}{\sigma_{ij}^2}\left\{\sum_{k=1}^{9} \Phi_k^{(ij)} w_k \alpha_k\right\}, \end{aligned} \tag{12}$$

where

$$\Phi_k^{(ij)} = \begin{cases} 1 & \text{if } k = [j - (2 - i)]\, r^{(2-i)} + [v - (i - 1)]\, r^{(i-1)} \\ & \text{for } v = 1, 2, 3, \text{ and } r = 3 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

By substituting (12) back into (11), we have

$$\Delta m_{ij}(t) = \eta_m \frac{2(z_i - m_{ij})}{\sigma_{ij}^2}\left\{\sum_{k=1}^{9} \Phi_k^{(ij)} \delta_k^{(3)} \alpha_k\right\} \quad \text{for } i = 1, 2,\ j = 1, 2, 3, \tag{14}$$

where $\delta_k^{(3)}$ is the error signal associated with the kth node in layer3 defined by

$$\delta_k^{(3)} = -\frac{\partial E}{\partial net_k^{(3)}} = \delta^{(4)} w_k. \tag{15}$$

The $\Phi_k^{(ij)}$ in (13) describes the connectivity of the nodes in layer3 which is associated with the ijth node in layer2. That is, if the kth node in layer3 has one of its incoming signal from the ijth node in layer2, then $\Phi_k^{(ij)} = 1$; otherwise $\Phi_k^{(ij)} = 0$.

TABLE I

THE USEFULNESS OF THE PROPOSED "P-CONTROLLER" COMPENSATOR.

Image	Original Entropy	Case 1: without online training area				Case 2: with 4-pixel online training area			
		B Absent (A1)	B Present (B1)	Difference (C1) = (A1) - (B1)	Percentage (C1)/(A1)	B Absent (A2)	B Present (B2)	Difference (C2) = (A2) - (B2)	Percentage (C2)/(A2)
Airplane	6.71	4.09	4.07	0.01	0.3%	4.09	4.08	0.01	0.3%
Baboon	7.36	6.11	6.10	0.01	0.1%	6.11	6.07	0.04	0.6%
Balloon	7.35	2.99	2.98	0.01	0.3%	2.97	2.96	0.01	0.4%
Barb	7.47	4.89	4.81	0.07	1.5%	4.83	4.72	0.11	2.3%
Barb2	7.48	4.90	4.87	0.03	0.6%	4.90	4.89	0.01	0.1%
Boats	7.10	4.23	4.17	0.06	1.4%	4.23	4.16	0.07	1.7%
Camera	7.01	4.86	4.85	0.01	0.2%	4.94	4.89	0.05	1.0%
Couple	6.39	4.12	4.07	0.05	1.2%	4.08	4.06	0.02	0.5%
Goldhill	7.53	4.64	4.64	0.00	0.0%	4.65	4.64	0.01	0.2%
Lena	7.59	4.67	4.67	-0.00	0.0%	4.67	4.68	-0.01	-0.2%
Lennagrey	7.45	4.32	4.32	-0.00	0.0%	4.33	4.34	-0.00	-0.1%
Noisesquare	5.72	5.44	5.43	0.01	0.2%	5.48	5.46	0.01	0.2%
Peppers	7.59	4.58	4.56	0.02	0.4%	4.55	4.53	0.01	0.3%
Shapes	6.74	2.55	2.48	0.07	2.7%	2.41	2.36	0.05	1.9%
Average	7.11	4.46	4.43	0.03	0.6%	4.44	4.42	0.03	0.6%

B-2. Primary part: Derivation for $\Delta \sigma_{ij}$**.**

The correction term for σ_{ij}, i.e., the standard deviation of the jth Gaussian membership function associated with the ith input variable, is given by

$$\Delta \sigma_{ij}(t) = \eta_\sigma (d - y)\left[\frac{\partial net^{(4)}}{\partial \sigma_{ij}}\right], \tag{16}$$

where η_σ is the learning rate, and $\frac{\partial net^{(4)}}{\partial \sigma_{ij}}$, by applying a similar approach for that of Δm_{ij}, can be expressed as

$$\frac{\partial net^{(4)}}{\partial \sigma_{ij}} = \frac{2(z_i - m_{ij})^2}{\sigma_{ij}^3}\left\{\sum_{k=1}^{9} \Phi_k^{(ij)} w_k \alpha_k\right\}. \tag{17}$$

By substituting (17) back into (16), we have

$$\Delta \sigma_{ij}(t) = \eta_\sigma \frac{2(z_i - m_{ij})^2}{\sigma_{ij}^3}\left\{\sum_{k=1}^{9} \Phi_k^{(ij)} \delta_k^{(3)} \alpha_k\right\} \quad \text{for } i = 1, 2, \text{ and } j = 1, 2, 3. \tag{18}$$

IV. EXPERIMENTS

In this section, performance of the proposed TS-FNN based lossless image coding system will be evaluated. For network adaptation process, the online training area is chosen to be the four nearest causal pixels $x_1, x_2, \ldots, x_4$ (Fig. 2) or even not used in this paper. Furthermore, the training cycle for network adaptation is set to be one in our experiments so that the run time performance of the proposed approach can be very good.

A. The "P-controller" based compensation mechanism.

The usefulness of the proposed "P-controller" compensator, i.e., the term $B_k u$ in (4), can be evaluated through the following experiment. In this experiment, the fourteen test images in Table I will be used as the predictor input. We first set the B matrix in the consequent part to be a zero vector (denoted as Type 1), and then calculate the first-order entropies of uncompensated prediction error for the fourteen test images. After that, the prediction process is repeated again but with the component $B_k u$ in existence this time (denoted as Type 2). Moreover, the experiment can be classified into two

TABLE II

COMPARISONS ON ACTUAL BIT RATES WITH EXISTING STATE-OF-THE-ART LOSSLESS IMAGE CODERS. (RUN ON A PENTIUM 1.4GHZ MACHINE)

Image	Proposed TS-FNN based coding system						JPEG-LS [2]	CALIC [1]	RALP [3]	TMW [4]
	Without online training area			With 4-pixel online training area						
	First Oreder Entropy	Bit rate	seconds	First Oreder Entropy	Bit rate	seconds				
Airplane	3.80	3.69	3.36	3.79	3.68	7.88	3.82	3.74	3.71	3.60
Baboon	6.02	6.00	3.97	6.00	5.97	8.31	6.04	5.88	5.81	5.73
Balloon	2.57	2.56	4.94	2.55	2.54	11.88	2.90	2.83	2.55	2.66
Barb	4.60	4.54	3.67	4.54	4.49	7.94	4.69	4.32	4.12	4.09
Barb2	4.68	4.62	5.72	4.71	4.66	15.92	4.69	4.53	4.51	4.38
Boats	3.91	3.80	5.36	3.90	3.78	12.27	3.93	3.83	3.75	3.61
Camera	4.61	4.40	1.00	4.61	4.39	2.05	4.31	4.19	4.24	4.10
Couple	3.76	3.73	1.03	3.75	3.73	2.31	3.70	3.61	3.63	3.45
Gold Hill	4.44	4.36	5.58	4.45	4.36	12.59	4.48	4.39	4.32	4.27
Lena	4.49	4.46	3.56	4.50	4.47	9.44	4.61	4.48	4.35	4.30
Lennagrey	4.11	4.07	3.52	4.12	4.08	10.08	4.24	4.11	3.95	3.91
Noisesquare	5.32	5.35	0.98	5.40	5.44	2.11	5.68	5.44	5.37	5.54
Peppers	4.38	4.35	3.55	4.35	4.33	8.00	4.51	4.42	4.27	4.25
Shapes	1.97	1.86	3.31	1.90	1.79	11.30	1.21	1.14	1.52	0.76
Average	4.19	4.13	3.54	4.18	4.12	8.72	4.20	4.07	4.01	3.90

cases, one with no online training area (denoted as Case 1), and the other with the predefined four-pixel online training region (denoted as Case 2).

In Table I, the results obtained by setting the B matrix to be a zero vector (Type 1) are shown in the columns denoted as A1 and A2, and that of obtained by using the proposed "P-controller" compensator (Type 2) are shown in the columns denoted as B1 and B2 respectively. For comparison purpose, we also show in the last column of each case (Table I) the percentage of improvement between the two types. As can be seen in Table I, the proposed "P-controller" compensator brings up a conspicuous improvement from 1.4% to 2.7% on the first-order entropy of the three images "Barb", "Boats", and "Shapes". That is, the proposed approach is very useful for images with many edges and lines.

B. Comparisons to existing state-of-the-art coders.

Table II gives actual bit rates by JPEG-LS [2], CALIC [1], RALP [3] and TMW [4] for a set of fourteen test images. The results of JPEG-LS, CALIC, RALP and TMW are taken directly from [3]. All the bit rates of the proposed algorithm are obtained using the same parameters with no individual optimization. Besides, the bit rates of the proposed approach are obtained with the "P-controller" compensator in presence. We also show in the second column and the fifth column respectively the first-order entropies of the compensated prediction errors using the the proposed approach. Moreover, the execution time of the proposed coder are also listed in the fourth and seventh column so that we can get a picture on the runtime performance. As can be seen in Table II, the proposed approach outperforms JPEG-LS [2] in almost all of the test images, and has lower bit rates than CALIC [1] in eight out of the fourteen test images. Encouragingly, the proposed approach achieves lower bit rates than the highly complex TMW in two images, "Balloon" and "Noise square".

C. Computational complexity of the proposed approach.

In this paper, the proposed TS-FNN based approach is designed to be practical and feasible under limited resources. Therefore, we set the training cycle to be one for each coding pixel such that the computational complexity for network adaptation can be reduced. Unlike other neural network based image predictors, the number of online training pixels is decreased substantially and even not used in the proposed approach so that the network adaptation process can be accelerated. As can be seen in previous sections, the major nonlinear operation in the proposed system is the calculation of the six Gaussian membership degree in the membership layer (2). In the network adaptation process, only addition, multiplication and division operations are needed. Therefore, a very good run time performance can be obtained (Table II). Actually, the computational complexity can be further reduced if we can define an error threshold so that the network adaptation process will not be activated until the prediction error is beyond the predefined threshold.

V. CONCLUSION

In this paper, we propose the use of a TS-FNN based predictor for lossless coding of images. For pixels around edges, we propose a novel approach by implementing implicitly the commonly used "P-controller" compensator in control system into the network weights so that the prediction result can be improved. It is noted that the proposed "P-controller" compensator is quite different to the so-called "bias cancelation" technique applied in most of the lossless image coders. As can be seen in our experiments, the use of the TS-FNN based predictor and the "P-controller" compensator render the proposed approach highly adaptable to the varying statistics of coding images. Besides, comparisons to existing state-of-the-art lossless image coders have demonstrated the usefulness of the proposed approach, and what's more, we have brought up an idea of enhancing the predictive coding efficiency in a quite different aspect.

REFERENCES

[1] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Communications*, Vol. 45, No. 4, pp. 437-444, April 1997.

[2] 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.

[3] L.-J. Kau and Y.-P. Lin, "Least Squares-Based Switching Structure for Lossless Image Coding," *IEEE Trans. Circuits and Systems I*, Vol. 54, No. 7, pp. 1529-1541, July 2007.

[4] B. Meyer and P. E. Tischer, "TMW-a new method for lossless image compression," in *Proc. Int. Picture Coding Symp.*, Berlin, Germany, Oct. 1997.

[5] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, Vol. 28, pp. 15-33, 1988.

[6] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man. Cyber.*, Vol. 15, pp. 116-132, 1985.

[7] C. H. Lee and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Trans. Fuzzy Systems*, Vol. 8, No. 4, pp. 349-366, 2000.