Postprocessing for Image Coding Applications Using Neural Network Visual Model

Z. He†, S. Chen, B. Luk and R. Istepanian

Department of Electrical and Electronic Engineering
University of Portsmouth, Anglesea Building
Portsmouth PO1 3DJ, U.K.
Tel/Fax: +44 (0)1705 842599/842561
Email: schen@ee.port.ac.uk

†Currently at: LTX Corporation, University Avenue, Westwood
MA 02090-2306, USA

Abstract

We present a neural network visual model (NNVM), which extracts multi-scale edge features from the decompressed image and uses these visual features as input to estimate and compensate the coding distortions. Our approach is a generic postprocessing technique and can be applied to all the main coding methods. Experimental results involving post-processing four coding systems show that the NNVM significantly improves the quality of reconstructed images, both in terms of the objective peak signal to noise ratio (PSNR) and subjective visual assessment.

1 Introduction

Image coding is always a trade-off between the coding bit rate and the coded image quality. Generally speaking, increasing coding bit rate can improve quality, but this is limited by channel bandwidth or storage capacity. Postprocessing offers an alternative to improve the quality of reconstructed image without increasing bit rate. Existing postprocessing methods [1]–[6] employ filtering to smooth blocking artifacts, and are designed for entertainment applications where obtaining pleasant viewing quality is the main purpose. Since filtering also

0-7803-5060-X/98/$10.00 © 1998 IEEE

causes oversmoothing on image edges, these methods are not appropriate for applications which require genuinely good image quality with minimum distortions, such as tele-medical diagnosis where contours of objects are crucial for correct diagnosis, or satellite remote sensing where original image edges are essential for proper classification and recognition. In addition, existing methods are specifically designed for fixed-block transform coding (TC) and vector quantization (VQ), and they cannot be applied to non-block predictive coding (PC) or variable-block quadtree (QT) coding.

We develop a generic postprocessing technique for all the major coding systems based on the direct correction of actual coding distortions. The key to the proposed technique is a NNVM for recovering the distortion image based on the decoded image. The distortion image is defined as the difference image between the original image and the decoded image. This distortion-recovery model consists of a visual or gradient feature extractor to extract edge information from the decoded image, and a one-hidden-layer neural network to estimate the distortion image using the visual features of the decoded image. We demonstrate the advantages of the proposed postprocesing technique on four coding systems, namely TC, VQ, QT coding and PC. Our experimental results show that the NNVM achieve significant improvements on the quality of reconstructed images, in both the objective distortion measure and subjective viewing evaluation.

2 The NNVM for postprocessing

The proposed approach is illustrated in Fig.1. It can be shown that main coding distortions are edge distortions, including blurred edges and blocking artifacts. Therefore, the main task of the model is to correct edge distortions, and we adopt the strategy depicted in Fig. 2. A decoded image of size $N \times N$ is divided into blocks of size $n \times n$, and pixels of each block are fed into a visual feature extractor, which extracts edge features of the block. These edge features are fed into a one-hidden-layer neural network, which produces an estimate of the corrresponding distortion image block. We will refer to $n \times n$ as the postprocessing block size. The basic idea is that, after learning, the

output of the model will be a good estimate of the distortion image, which can then be added to the decoded image to compensate actual coding distortions.

Edge features of an image block are extracted as multi-scale first-order derivatives. To calculate derivatives for an $n \times n$ block in different scales, the block is recursively divided into 4 equal-size sub-blocks until the sub-block size is reduced to 2×2. For a generic sub-block X_s of size $n_s \times n_s$, a pair of horizontal and vertical derivatives (d_h, d_v) are calculated as:

$$\begin{aligned} d_h &= \sum_{i=1}^{n_s} \sum_{j=1}^{n_s/2} X_s(i,j) - \sum_{i=1}^{n_s} \sum_{j=1+n_s/2}^{n_s} X_s(i,j), \\ d_v &= \sum_{i=1}^{n_s/2} \sum_{j=1}^{n_s} X_s(i,j) - \sum_{i=1+n_s/2}^{n_s} \sum_{j=1}^{n_s} X_s(i,j) \end{aligned} \tag{1}$$

where $X_s(i,j)$ is the pixel value at position (i,j) in X_s. The outputs of the visual feature extractor, the multi-scale derivatives, can be arranged in a vector form: $\mathbf{d} = [d_1 \; d_2 \; \cdots d_M]^T$. The total number of derivatives, M, for an $n \times n$ block is determined by the formula

$$M = 2 \sum_{i=1}^{\log_2 n} \left(\frac{n}{2^i} \right)^2 \tag{2}$$

The choice of the block size $n \times n$ has important influence on the complexity and performance of the model. Ideally, the block size should be as large as possible. However, too large a block size would make computation and storage impractical. For post-processing block-based coding systems, the postprocessing block size should be larger than the coding block size, so that blocking artifacts at coding block boundaries can be corrected.

The inputs to the one-hidden-layer neural network, the edge features, are normalized to lie in the range of $(-1, 1)$, the hidden-layer outputs of the NNVM are given by

$$h_k = f\left(\sum_{l=1}^{M} V_{l,k} \cdot d_l + V_{0,k} \right), \quad 1 \le k \le H_n, \tag{3}$$

where H_n is the number of hidden neurons, and the outputs of the NNVM are given by

$$\hat{Y}(i,j) = \alpha \cdot f\left(\sum_{k=1}^{H_n} W_k(i,j) \cdot h_k + W_0(i,j)\right) + \beta, \quad 1 \leq i,j \leq n \quad (4)$$

where α and β are fixed scaling and shifting constants for mapping the model outputs onto the range of pixel values. The activation function f is the usual bipolar sigmoid function. The number of the hidden-layer neurons, H_n, is determined during training by starting with a small hidden layer and gradually increasing the hidden layer size until the performance stops improving. The total number of adjustable parameters, P_{NNVM}, for the NNVM is

$$P_{NNVM} = n \times n \times (H_n + 1) + H_n \times \left(1 + 2\sum_{i=1}^{\log_2 n}\left(\frac{n}{2^i}\right)^2\right) \quad (5)$$

To collect training data from a coding system, an $N \times N$ training image is compressed and then decompressed. The corresponding distortion image is obtained by subtracting the decoded image from the original image. The decoded and distortion images are divided into $n \times n$ blocks. A pair of blocks gives rise to a pair of input and desired output. As an image can only provide $\frac{N \times N}{n \times n}$ pairs of training data, many images should be used to collect sufficient training data samples. The network weights $V_{l,k}$ and $W_k(i,j)$ are learnt using the backpropagation algorithm.

3 Experimental results

The proposed postprocessing technique was applied to TC, VQ, QT coding and PC systems. The coding algorithms were the JPEG [7] for TC, the algorithm based on the Kohonen self-organizing feature map [8] for VQ, the improved QT algorithm [9] for QT coding, and the algorithm using a neural network predictor [10] for PC. Sixteen images of size 512×512 with 8 bits per pixel (bpp) were involved in the experiment. The images, "peppers", "airplane", "goldhill", "lake", "announcer", "cornfield", "windows" and "yacht", were used to provide training data, and the other images, "lena", "littlegirl",

"zelda", "boats", "cablecar", "hatgirl, "kids" and "soccer", were used as test images. We also implemented two typical existing postprocessing methods, Reeve's filtering method [1] and Paek's algorithm [4], to compare them with our approach in the identical TC and VQ environments. It should be pointed out that, like most of the existing methods, these two algorithms are impractical for post-processing QT and PC systems.

An adequate postprocessing block size was found by the experiments to be 16×16 for JPEG, VQ and QT coding, and 8×8 for PC. The experimental results also suggested that $H_n = 40$ was sufficient. Tables 1 and 2 compare the postprocessing gains obtained using the NNVM and two existing algorithms for the JPEG and VQ. Tables 3 and 4 list the postprocessing gains obtained by the NNVM for the QT coding and PC. Fig.3 depicts the original and VQ coded images of "Lena" together with the three post-improved images. These pictures give magnified portions of the images for a clearer visual evaluation. The results demonstrate that the NNVM has superior performance over Reeve's and Paek's algorithms for post-processing TC and VQ systems. The performance achieved by the NNVM in this study is generally better than the existing postprocessing methods. For example, a sophisticated space-variant filtering method [2] only achieved a PSNR gain of 0.40 dB for a VQ coded "Lena" image at an original coding PSNR of 29.90 dB. The NNVM achieved a postprocessing gain of 0.80 dB for a VQ coded "Lena" image at an original coding PSNR of 30.20 dB. The results also confirm that the NNVM is very effective for post-improving QT coding and PC systems. In contrast, most of the existing postprocessing algorithms cannot be applied to these two coding systems.

4 Conclusions

A generic postprocessing technique for image coding has been developed based on a NNVM. Unlike most existing postprocessing methods which basically smooth blocking artifacts to achieve better viewing quality, the proposed technique corrects actual coding losses. As a result, our method is applicable to all of the major coding methods

while existing postprocessing methods are limited to TC or VQ. Experiments of applying the proposed technique to four coding systems have been conducted, and the results obtained confirm that the proposed technique has much better postprocessing gains and wider applications over existing methods.

References

[1] H.C. Reeve and J.S. Lim, "Reduction of blocking effect in image coding," in *Proc. ICASSP* (Boston, USA), 1983, pp.1212–1215.

[2] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.ASSP-34, pp.1258–1267, 1986.

[3] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.2, pp.91–95, 1992.

[4] H. Paek, J.W. Park and S.U. Lee, "Non-iterative post-processing technique for transform coded image sequence," in *Proc. ICIP*, 1995, pp.208–211.

[5] C. Derviaux, F.X. Coudoux, M.G. Gazalet, P. Corlay and M. Gharbi, "A postprocessing technique for block effect elimination using a perceptual distortion measure," in *Proc. ICASSP*, 1997, pp.3001–3004.

[6] J.D. McDonnell, R.N. Shorten and A.D. Fagan, "Postprocessing of transform coded images via histogram-based edge classification," *J. Electronic Imaging*, Vol.6, pp.114–124, 1997.

[7] W.B. Pennebaker and J.L. Mitchell, *JPEG Still Image Data Compression Standard.* New York: Van Nostrand Reinhold, 1993.

[8] N.M. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organizing feature maps," in *Proc. ICNN*, 1988, pp.101–105.

[9] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE Trans. Image Processing*, Vol.3, pp.207–215, 1994.

[10] S.A. Dianat, N.M. Nasrabadi and S. Venkataraman, "A non-linear predictor for differential pulse-code encoder (DPCM) using artificial neural networks," in *Proc. ICASSP*, 1991, pp.2793–2796.

Coding image	JPEG coded PSNR (dB)	postprocessing gain (dB)		
		NNVM	Reeve's	Paek's
JPEG: Quality=7				
Lena	28.85	0.81	0.69	0.36
Littlegirl	29.04	0.79	0.77	0.26
Zelda	29.98	0.80	0.73	0.68
Boats	28.23	0.81	0.52	0.28
Cablecar	27.84	0.79	0.69	0.16
Hatgirl	30.60	0.75	0.72	0.47
Kids	28.19	0.69	0.68	0.24
Soccer	27.34	0.73	0.64	0.21
JPEG: Quality=14				
Lena	31.67	0.71	0.30	0.15
Littlegirl	32.26	0.69	0.59	0.08
Zelda	33.22	0.55	0.38	0.19
Boats	31.01	0.77	0.07	0.13
Cablecar	30.96	0.71	0.23	0.07
Hatgirl	34.15	0.58	0.21	0.24
Kids	31.56	0.64	0.21	0.05
Soccer	30.76	0.68	0.54	0.05

Table 1: PSNR values of JPEG coding and postprocessing gains.

Coding image	VQ coded PSNR (dB)	postprocessing gain (dB)		
		NNVM	Reeve's	Paek's
VQ: bit rate=0.25 bpp				
Lena	26.53	1.13	0.64	0.02
Littlegirl	27.34	1.10	0.69	0.06
Zelda	28.79	1.01	0.67	0.11
Boats	25.19	0.97	0.45	0.07
Cablecar	24.51	0.95	0.33	0.06
Hatgirl	27.37	1.17	0.61	0.03
Kids	25.34	0.93	0.37	0.05
Soccer	23.85	1.19	0.56	0.08
VQ: bit rate=0.5 bpp				
Lena	30.20	0.80	0.25	0.01
Littlegirl	31.50	0.83	0.43	0.02
Zelda	32.49	0.81	0.35	0.01
Boats	29.74	0.76	0.10	-0.01
Cablecar	28.72	0.58	0.06	0.00
Hatgirl	32.51	0.76	0.17	0.00
Kids	29.73	0.65	0.03	0.00
Soccer	28.65	0.78	0.38	0.01

Table 2: PSNR values of VQ coding and postprocessing gains.

Coding image	QT: bit rate=0.25 bpp		QT: bit rate=0.5 bpp	
	QT coded PSNR (dB)	gain (dB) NNVM	QT coded PSNR (dB)	gain (dB) NNVM
Lena	29.66	0.91	32.39	0.84
Littlegirl	29.87	0.96	32.42	1.01
Zelda	31.38	1.00	33.90	0.89
Boats	28.72	0.85	31.80	0.77
Cablecar	27.83	0.91	30.59	0.93
Hatgirl	33.29	1.14	36.86	0.80
Kids	28.38	0.92	31.14	0.98
Soccer	25.83	1.11	28.32	1.23

Table 3: PSNR values of QT coding and postprocessing gains.

Coding image	PC: coding bit=1		PC: coding bit=2	
	PC coded PSNR (dB)	gain (dB) NNVM	PC coded PSNR (dB)	gain (dB) NNVM
Lena	24.17	2.87	28.42	1.43
Littlegirl	27.51	2.01	34.07	0.51
Zelda	29.59	1.20	34.79	0.38
Boats	23.84	3.42	28.39	2.32
Cablecar	22.38	2.88	25.95	2.39
Hatgirl	23.71	3.52	28.42	2.84
Kids	23.88	2.66	27.83	1.95
Soccer	21.88	2.97	26.03	2.05

Table 4: PSNR values of PC (quantizing step=4) and postprocessing gains.

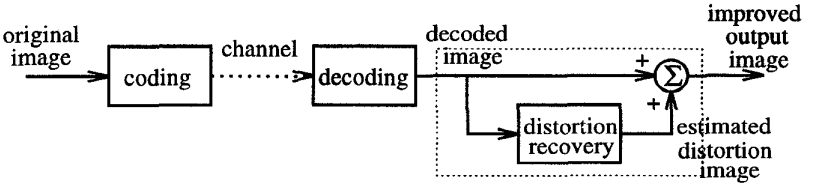


Figure 1: A generic postprocessing approach for image coding.

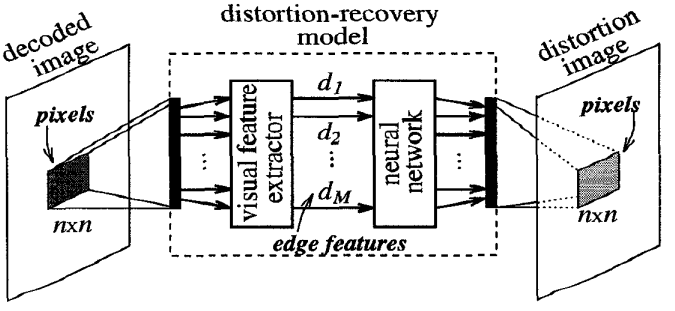
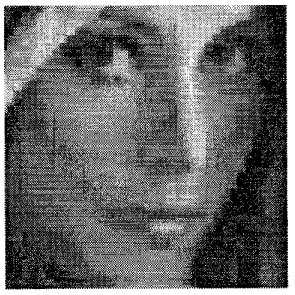


Figure 2: Schematic of the neural network visual model.

(a) Original image

(b) VQ coded (26.53 dB)



(c) NNVM (gain 1.13 dB)

(d) Reeve's (gain 0.64 dB)

(d) Paek's (gain 0.02 dB)

Figure 3: Magnified portions of original, VQ coded (bit rate=0.25 bpp) and post-improved images of "Lena."