Improved Image Compression using Backpropagation Networks

G. Qiu, T. J. Terrell and M.R. Varley

Department of Electrical and Electronic Engineering, University of Central Lancashire, U.K.

Abstract

This paper describes an improved image compression scheme using backpropagation networks. The new scheme is aimed at improving the networks' generalisation capabilities, thereby enabling them to effectively compress a wide range of novel images. The networks operate, and are trained on, residual image blocks, thus eliminating the problem of varying average image intensities highlighted by Cottrell et al. [3].

Tabulated experimental results and example reconstructed novel images, for the method of [3] and the new technique, are presented, which demonstrate the improved image compression performance gained using this new technique.

1: Introduction

The backpropagation technique described by Rumelhart *et al.* [1] was a very significant development in the field of neural networks, and has found many applications in a wide range of areas of research [2]. The application of a simple three-layer backpropagation network for image data compression was first proposed by Cottrell *et al.* [3] and subsequently studied and developed by others [4,5]. Some inherent features of backpropagation network image data compression schemes are: (a) the network structure is massively parallel, (b) the network is adaptive, (c) the network determines the compressed features of the original image in a self-organising manner during the training stage, and (d) the intrinsic generalisation property of the structure enables it to process images outside the training set (novel images) effectively.

It has been reported that image compression schemes based on backpropagation networks can achieve a performance comparable to other existing image compression schemes [3,4,5]. Several problems with image data compression using backpropagation networks have been revealed by these initial studies. One main problem is that the performance of the system degrades when used to compress images that are not statistically consistent with those used for training. Thus a primary objective is to improve the generalisation capabilities of the system, hence enhancing its performance for a wide range of novel images. Cottrell *et al.* [3] pointed out that one source of error in the reconstructed novel images is due to the fact that their average intensities differ from those of the training images. Although a number of researchers [4,5] have studied and commented on the backpropagation image compression scheme of Cottrell *et al.*, this problem has not yet been fully addressed.

Motivated by studies of the backpropagation training algorithm [6] and the applications of backpropagation networks to image compression [3], we developed an improved image compression scheme, operating on residual blocks. Experimental results are presented which show that this new scheme exhibits enhanced generalisation capabilities, and it is shown that performance is improved in terms of the mean square error and the visual quality of the reconstructed images.

2: Image Compression using a Backpropagation Network

In this section, we briefly review the basic idea of using a backpropagation network to achieve image compression. A number of researchers [3,7,8] have shown that multilayer perceptron networks are able to

0-8186-5845-2/94 $3.00 © 1994 IEEE

learn a transform for reducing signal redundancy, and are capable of learning a reverse transform to recover the information (with some degradation) from a more compact (compressed) form of representation.

The network shown in Fig. 1 has N input nodes, H hidden layer nodes ($H<N$) and N output nodes. The input to the network, X, is a vector of dimension N, and in our application, X is an $m \times n$ ($N = m \times n$) block of pixels extracted from an image. The sum of the inputs, $s_h(i)$, for node i in the hidden layer is calculated as

$$s_h(i) = \sum_{j=1}^{N} w_h(j,i)x(j) + b_h(i) \quad \text{for } 1 \le i \le H \qquad (1)$$

where $w_h(j,i)$ is the connection weight from jth input node to the ith node in the hidden layer, $x(j)$ is the jth input element (pixel value), and $b_h(i)$ is the bias input of the ith hidden layer node.

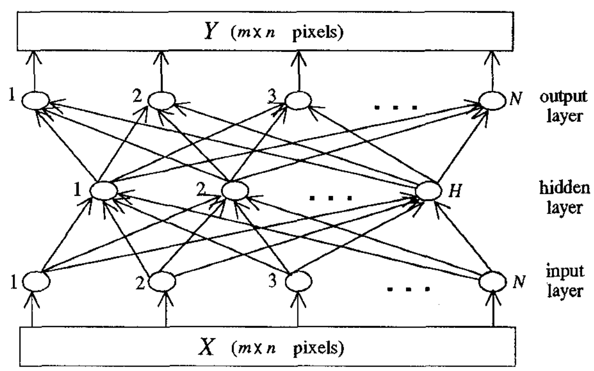

Fig. 1 Three-layer Backpropagation Network for Image Data Compression

Similarly, the sum of the inputs to the ith output node is calculated as

$$s_o(i) = \sum_{j=1}^{H} w_o(j,i)h(j) + b_o(i) \quad \text{for } 1 \le i \le N \qquad (2)$$

where $w_o(j,i)$ is the connection weight from the jth hidden layer node to the ith output node, $b_o(i)$ is the bias input to the ith output node, and $h(j)$ is the output of the jth hidden layer node. A sigmoid function defined by equation (3) is applied to $s_h(i)$ and $s_o(i)$ to obtain the output value of the node for the hidden and output layers respectively.

$$f(s) = \frac{1-e^{-s}}{1+e^{-s}} \qquad (3)$$

where s is the value of the sum of the inputs to a node in the layer (hidden or output).

In order to relate the grey levels of the images (usually in the range 0 to 255) to the values produced by this function, the grey levels of the images were converted linearly from the original range to the range 0 to 1 [3].

To train the network, X is used as the input and as the desired output, and the backpropagation training algorithm [1] is applied. The squared error function, minimised during training, is defined as

$$E = \frac{1}{2}\sum_{i=1}^{N}\left(x(i) - f(s_o(i))\right)^2 \qquad (4)$$

After training, the network is ready for operational use. The image to be compressed is partitioned into contiguous, non-overlapped blocks which are presented to the network one at a time. The outputs of the hidden layer nodes constitute the compressed features of an input block. To achieve compression in a practical sense, the outputs of the hidden layer nodes are quantized. Because the sigmoid function (equation (3)) forces all hidden layer node outputs into the range -1 to 1, a simple quantization scheme can be adopted. For the results presented in this paper, a scalar quantizer was used for this purpose.

3: Centralised Backpropagation (CBP) Network for Image Compression

The backpropagation algorithm has a number of drawbacks, including a slow rate of convergence and a dependence on the initial settings of the weights. Although work has been done by other researchers to improve the basic algorithm, we investigated one particular modification, originally proposed by Stornetta and Huberman [6]. They showed that by setting a symmetric dynamic range for input, hidden and output layer nodes, the learning rate of the basic backpropagation network could be significantly improved. It was shown that the use of such a symmetric structure also improved the uniformity of training, i.e. the network was less sensitive to the initial values of the weights. Taking these points further, we developed the centralised backpropagation (CBP) network structure for image compression applications.

3.1: Use of Residual Blocks

Recalling that one of the sources of error in the reconstruction of novel images is that their average intensities differ from those of the training images, a modification to the basic compression scheme described in section 2 may be introduced, whereby the block mean value is removed prior to training. The block mean value M, defined in equation (5), is coded separately and added back to each pixel value during reconstruction.

$$M = \frac{1}{N}\sum_{i=1}^{N} x(i) \qquad (5)$$

where N is the number of pixels in the block.

Mean removal is a common technique for other block based compression methods, such as vector quantization [9] and the Karhunen-Loeve Transform [10]. However, the motivation for mean removal when using a backpropagation network for image compression is to improve the training of the network and to enhance its generalisation capabilities to deal effectively with a wide range of novel images.

When applying the backpropagation network to perform image compression, the network should operate on residual image blocks. Using residual blocks improves the generalisation capabilities of the backpropagation network by removing the effect of the different average intensities between training images and novel images, since all residual blocks have the same average intensity of zero. The reconstruction errors arising due to the different average intensities of the training and novel images [3] will thus be eliminated.

It was suggested in [6] that training will be improved if the input and desired output signals are symmetrical about zero; it is therefore appropriate to use the network to operate on residual image blocks which are inherently symmetrical about zero. If the original pixel intensity range is from 0 to 255, the residual image will have a pixel intensity range from -255 to 255, and by using appropriate scaling we may use a symmetrical network structure with a dynamic range of -1 to 1 for the input, hidden and output layer nodes.

Fig. 2 shows a comparison of the pixel histograms of two images (Fig. 2(a)) and their residual images (Fig. 2(b)). The residual image is obtained by subtracting the means in each contiguous non-overlapped 8x8 block. It is seen that in the original images, the grey level distributions of the two images differ quite significantly, whilst for the residual images, they are both centred about zero. Most of the residual image pixels fall in the range of -8 to 8. This was found to be generally true for a variety of different images.

The operation given by equation (6) tends to convert the original image pixel values into a set of numbers having more structure (less randomness). That is, the residual image blocks produced by equation (6) have similar first-order statistics. This is important in backpropagation network image compression applications because less randomness in input data means that training demands are reduced, and the generalisation capability of the compression scheme is improved.

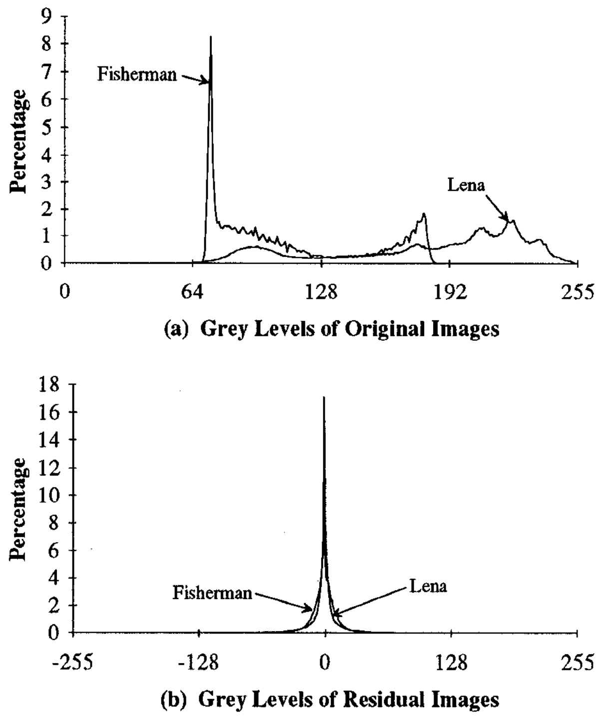

Fig. 2 Pixel Histograms of Original and Residual Images

An $m \times n$ block of image samples, $X = \{x(1), x(2), \cdots, x(N)\}$, where $N = m \times n$, has a block average intensity M, calculated according to equation (5). Its corresponding residual block $R = \{r(1), r(2), \cdots, r(N)\}$ is formed by subtracting the quantized block average intensity M_Q from the pixel values, i.e.

$$r(j) = x(j) - M_Q \quad \text{for } j = 1,2,\cdots,N \qquad (6)$$

If the original image has an intensity range of 0 to 255, then the residual image will have an intensity range of -255 to 255, and the average intensity of the residual image is zero. When the backpropagation network is used to code the residual image, the network has a symmetric structure similar to that described in [6]. The residual image pixels are converted linearly from the range of -255 to 255 to the range of -1 to 1, and thus the input, hidden and output layer nodes of the network will have a dynamic range of -1 to 1, i.e. the network is said to be 'centralised'.

In the training stage, the chosen training images are sub-divided into $m \times n$ blocks, the block means are calculated according to equation (5), and the residual blocks are formed using equation (6). These residual blocks constitute the training patterns for the network. In the operational coding stage, the image to be coded is sub-divided into $m \times n$ blocks. Each residual block is passed through the trained network, and the block mean is coded separately using a scalar quantizer. This quantized block mean and the quantized hidden node outputs of the network constitute the compressed data for the block.

4: Experimental Results

In this section, we present image data compression results using the backpropagation techniques described in this paper. The block size used was 8x8 pixels. The training samples were 8x8 non-overlapped blocks extracted from two 256x256 images (Jazz (Fig. 3(a)) and Fisherman (Fig. 3(b))) arbitrarily selected from our image library, each with a pixel intensity range of 0 to 255. Four novel images (F-16, Peppers, Sailboat and Lena) were used to test the system. We demonstrate that the proposed technique improves the networks' generalisation capabilities as compared with the original scheme [3]. It should be noted that all residual pixels were found to lie in the range of -64 to 64, and hence a scale factor of 1/64 was used to convert the residual pixel values to the range of -1 to 1.

Table 1 shows a comparison of the peak signal-to-noise ratio (PSNR) performance, calculated according to equation (7), for our new method and the method of Cottrell *et al.* (the chosen benchmark). In all cases, the networks were trained for 500 iterations. For our new method, the block mean was quantized using an 8-bit scalar quantizer and all the hidden node outputs were also quantized to 8-bit resolution. It can be seen that for images inside and outside the training set, the new method provides improved PSNR performance. The improvements achieved for the training images suggest that our new method can train the network (to a given required performance level) faster, and the improvements achieved for the novel images indicate that the generalisation capability of our system is significantly better than that of the method described in [3]. It should be noted that for our method, the block mean must be coded, and we therefore use one less hidden layer node to achieve the same compression ratio as method [3].

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{L \times 255^2}{\sum_{j=1}^{L} \left(x(j) - x_r(j)\right)^2} \right) \qquad (7)$$

where L is the total number of pixels in the image, and $x(j)$ and $x_r(j)$ are the pixel values of the original and reconstructed images respectively.

Figures 4-7 show the reconstructed images using the method of [3] and our new method, for the four novel images for a selected bit rate of 1 bit per pixel (bpp). Also shown are the difference images between the original and the reconstructed images using the two methods, where the pixel difference values are multiplied by 8 to enable a satisfactory visual presentation of comparative performance.

It is seen from these images that the new centralised backpropagation method performs very well in the low detail areas of the image, whilst suffering some degradation in high detail areas. However this is seen to be much less significant when compared with the method of [3], where not only the high detail areas but also the low detail areas are degraded.

5: Concluding Remarks

In this paper we have presented an improved technique for image data compression using backpropagation networks. It has been shown that the scheme of [3] was significantly improved by our modifications. Particularly, it can be seen that for the four novel images (F-16, Peppers, Sailboat and Lena), the new scheme has significantly improved compression performance due to its enhanced generalisation capabilities. It should be pointed out that, in the experiments, no attempt was made to optimise the training parameters for the networks, and investigations into network size and quantization resolution for hidden node outputs were not conducted. These points are being addressed by the authors as part of the continuing development work in this area of research. It should also be noted that this work does not compare image compression performance with results obtained from established techniques such as the Discrete Cosine Transform (DCT), but instead is aimed at improving the performance of the backpropagation network image compression scheme.

6: References

[1] Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 'Learning internal representations by error propagation', in *Parallel distributed processing: Explorations in the microstructure of cognition*, D. E. Rumelhart and J.L. McClelland (Eds.), MIT Press,1986

[2] Roth, M.W., 'Neural - network technology and its applications', Johns Hopkins APL Technical Digest, Vol. 9, No. 3, 1988

[3] Cottrell, G.W., Munro, P. and Zipser, D., 'Image compression by backpropagation: An example of extensional programming', in *Models of cognition: a review of cognition science*, pp. 208-240, N. E. Sharkey, (Ed.), Norwood,1989

[4] Sonehara, N., Kawato, M., Miyake, S. and Nakane, N., 'Image data compression using a neural network model', in *Proc. International Joint Conference on Neural Networks*, Washington DC, pp. II-35-II-41, 1989

[5] Marsi, S., Ramponi, G. and Sicuranza, G.L., 'Improved neural structures for image compression', in *Proc. ICASSP 91*, Vol.4, pp. 2821-2824, 1991

[6] Stornetta, W. and Huberman, B.A., 'An improved three-layer backpropagation algorithm', in *Proc. 1st Int. Conf. on Neural Networks*, pp.II-637-II-643, 1987

[7] Saund, E., 'Dimensionality-reduction using connectionist networks', in *IEEE Trans. PAMI*, Vol.11, pp. 304-314, 1989

[8] Cottrell, G.W. and Munro, P., 'Principal components analysis of images via backpropagation', in *SPIE Vol. 1001 Visual Communications and Image Processing '88*, pp. 1071-1077, 1988

[9] Baker, R.L. and Gray, R.M., 'Image compression using non-adaptive spatial vector quantization', in *Proc. 16th Asilomar Conf. on Circuit Syst Compt*, pp. 55-61, 1982

[10] Netravali, A.N. and Haskell, B.G., *Digital pictures, representation and compression*, Plenum Press, London 1989

		2 bits per pixel (bpp)		1 bit per pixel (bpp)	
		Method of [3] $N = 64, H = 16$	CBP $N = 64, H = 15$	Method of [3] $N = 64, H = 8$	CBP $N = 64, H = 7$
Training Images	JAZZ	35.50 dB	36.01 dB	32.08 dB	32.35 dB
	FISHERMAN	32.84 dB	33.07 dB	30.10 dB	30.26 dB
Novel Images	F-16	29.44 dB	33.57 dB	26.91 dB	29.95 dB
	PEPPERS	32.34 dB	33.79 dB	29.87 dB	31.01 dB
	SAILBOAT	29.88 dB	31.71 dB	27.24 dB	28.54 dB
	LENA	24.60 dB	34.95 dB	23.20 dB	31.89 dB

Table 1 PSNR Values for Reconstructed Images (2 bpp and 1 bpp)

(a) Jazz *(b) Fisherman*

Fig. 3 Training Images

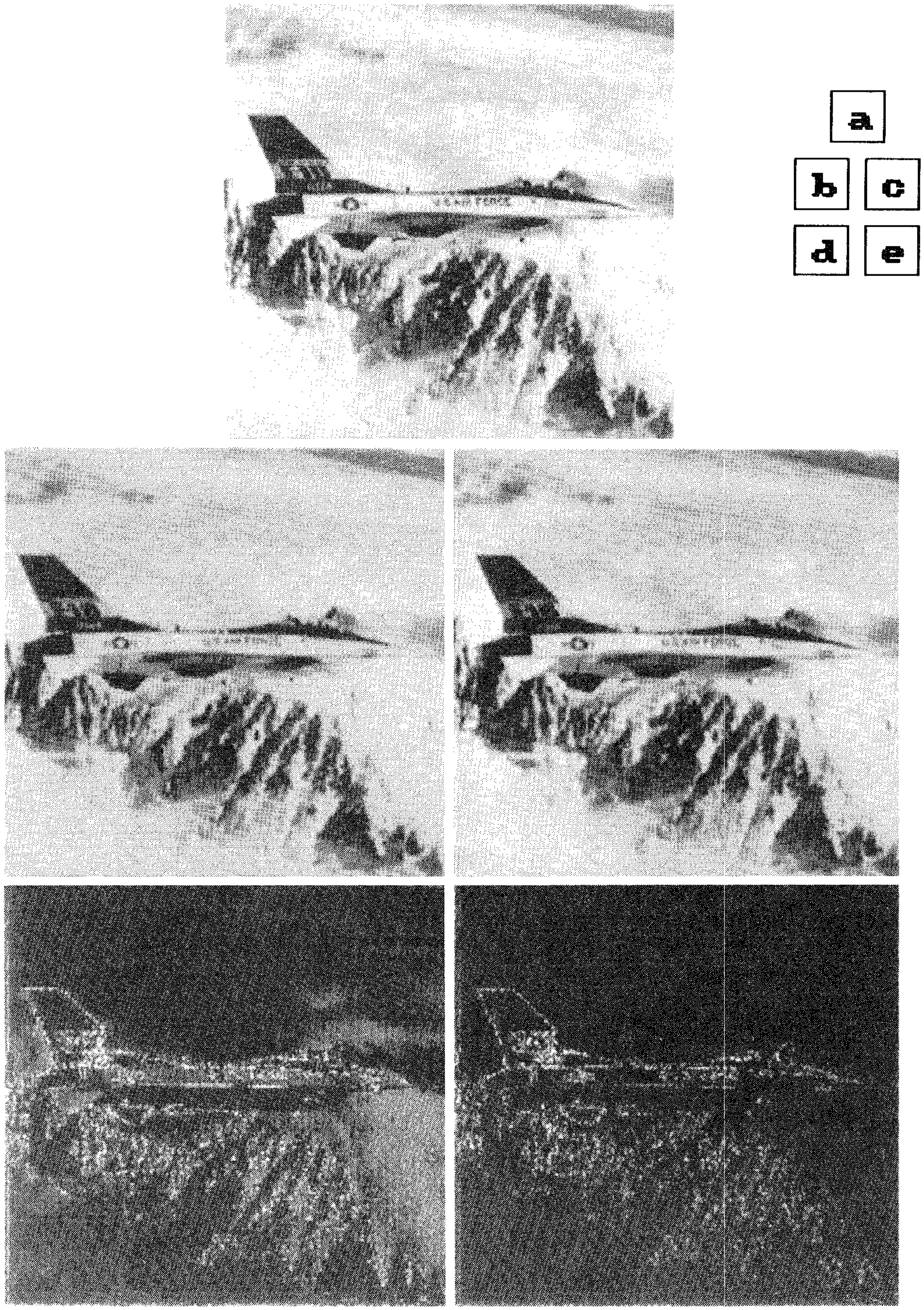

Fig. 4 F-16 Image
(a) Original Image
(b) Reconstructed Image using Method of [3]
(c) Reconstructed Image using Authors' CBP Method
(d) Difference Image between (a) and (b)
(e) Difference Image between (a) and (c)

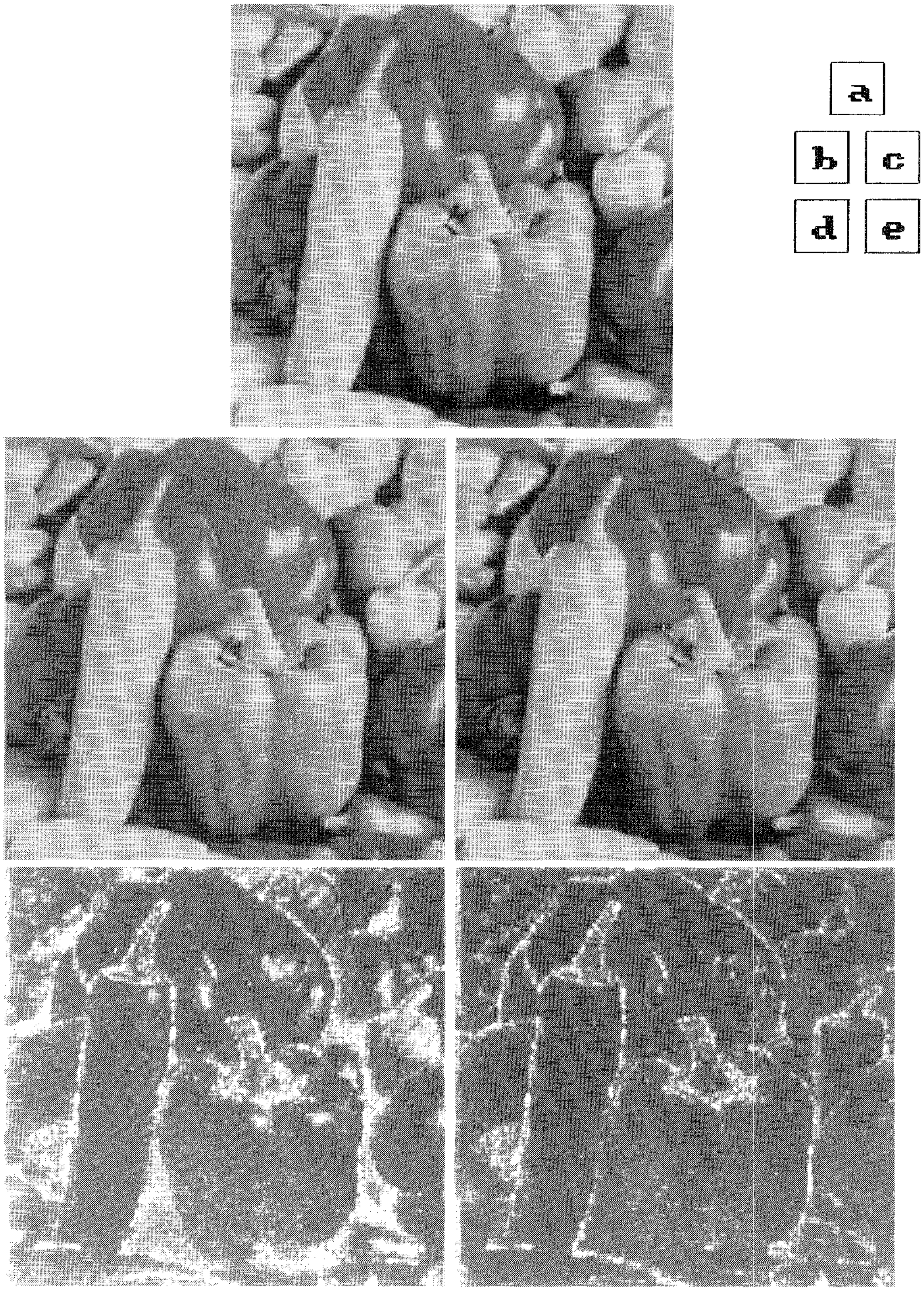

Fig. 5 Peppers Image
(a) Original Image
(b) Reconstructed Image using Method of [3]
(c) Reconstructed Image using Authors' CBP Method
(d) Difference Image between (a) and (b)
(e) Difference Image between (a) and (c)

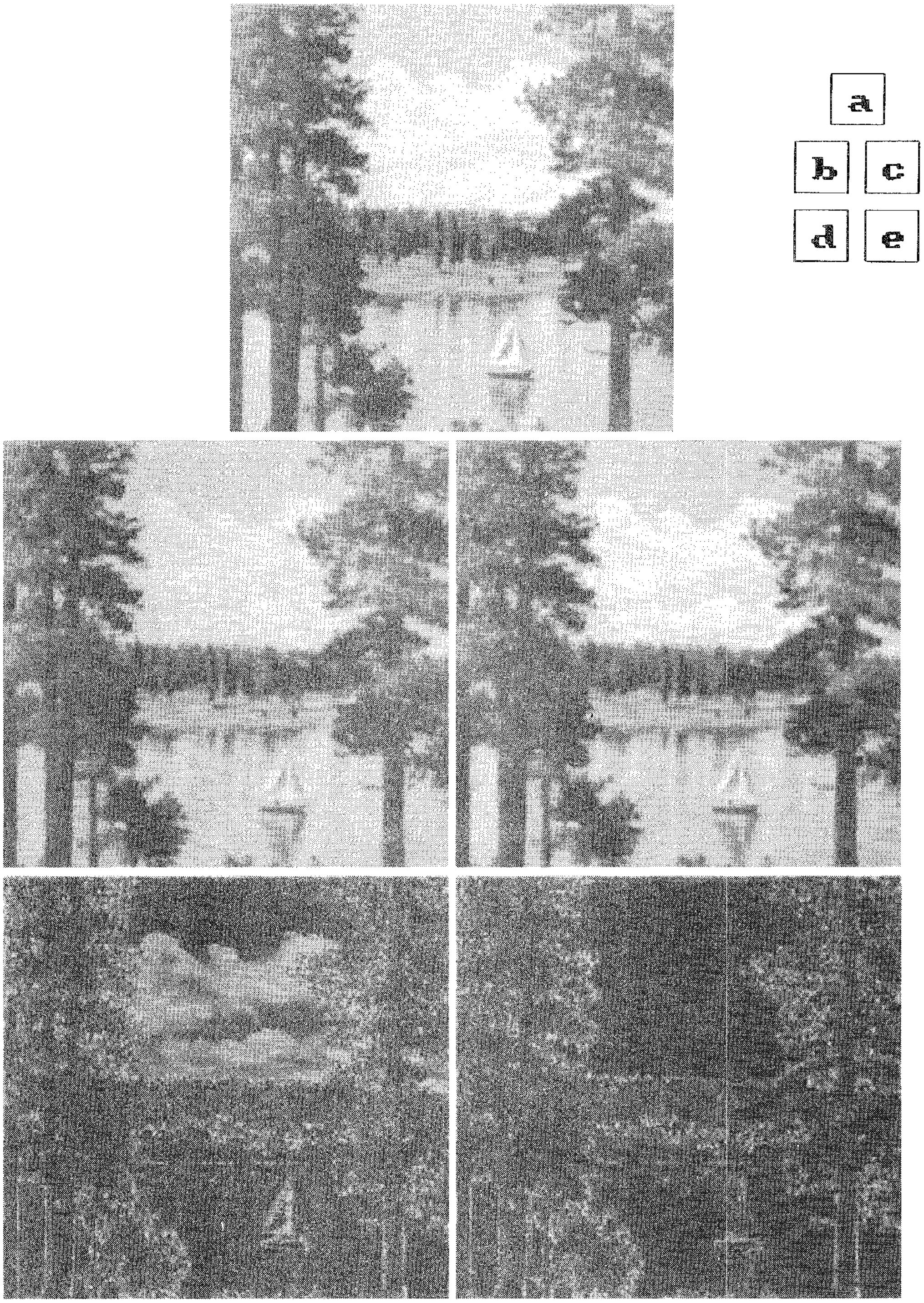

Fig. 6 Sailboat Image
(a) Original Image
(b) Reconstructed Image using Method of [3]
(c) Reconstructed Image using Authors' CBP Method
(d) Difference Image between (a) and (b)
(e) Difference Image between (a) and (c)

Fig. 7 Lena Image
(a) Original Image
(b) Reconstructed Image using Method of [3]
(c) Reconstructed Image using Authors' CBP Method
(d) Difference Image between (a) and (b)
(e) Difference Image between (a) and (c)