VARIABLE BIT RATE BLOCK TRUNCATION CODING FOR IMAGE COMPRESSION USING HOPFIELD NEURAL NETWORKS

G. Qiu, M. R. Varley and T. J. Terrell
University of Central Lancashire, UK

Abstract A Hopfield neural network based block truncation coding (BTC) technique is presented in this paper. For this scheme, BTC is formulated as the minimization of a cost function in which the bit map distributions for the blocks are explicitly included. It is explained that this cost function may also be interpreted as a measure of the block detail. Based on the observation of the final value of the cost function found by the Hopfield network, a block may be classified as a high detail block or a low detail block, which are coded differently, giving a different compression ratio for each type. It is shown that using this new technique, compression ratios up to 7:1 with good reconstructed image quality can be achieved. Experimental results are presented to demonstrate the effectiveness of this new scheme.

INTRODUCTION

Block Truncation Coding (BTC) for image compression [1] is a simple and effective coding technique, which compresses small sub-image blocks independently. In its original form, BTC is designed in such a manner that the reconstructed blocks retain the first and second moments of the original blocks. Compared with other block based compression methods, such as transform coding [2] and vector quantization [3], BTC has the advantage of being easy to implement and has the ability to code images with high visual quality and relatively high compression ratios.

In the original version of BTC, the mean value of the block is used as a threshold for classifying pixels into one of two classes. In our scheme, the computational power of the Hopfield neural network is employed to classify the pixels in a block into two classes. For the Hopfield network BTC there is no need to choose a threshold for the block - instead the coding problem is formulated as minimization of a cost function where the bit map distribution is explicitly included. Furthermore, this cost function is also used as a measure of visually significant attributes of the block. Based on the observation of the minimum value of this cost function computed by the Hopfield network, a new variation of the BTC technique is introduced with the aim of increasing compression ratio without sacrificing much of the visual quality of the reconstructed images.

Experimental results which demonstrate the effectiveness of the proposed Hopfield neural network based BTC technique are presented. Using the new technique, compression ratios of about 7:1 with good reconstructed image quality have been achieved.

BLOCK TRUNCATION CODING

For block truncation coding in its original form [1], the images to be compressed are divided into blocks. The pixels in each block, $x(1), x(2), \ldots, x(N)$, (where $N = n \times n$ for a block of size $n$) are individually quantized into two-level outputs in such a way that the first and second moments of the original block are preserved in the reconstructed block. The two-level quantizer threshold is given by

$$X_{tr} = \bar{X}$$  \hspace{1cm} (1)

and the quantizer output levels are

$$A = \bar{X} - \sigma \sqrt{\frac{q}{N - q}} \quad x(i) < X_{tr}$$ \hspace{1cm} (2a)

$$B = \bar{X} + \sigma \sqrt{\frac{N - q}{q}} \quad x(i) \geq X_{tr}$$

where $q$ is the number of pixels greater than or equal to $X_{tr}$, $\bar{X}$ is the block mean value, and $\sigma$ is the standard deviation of the block pixel values.

HOPFIELD NETWORK MODEL

Hopfield [4, 5, 6] has shown that neurons can be highly and selectively interconnected to give rise to collective computational properties and create a network with high computational efficiency. Collective computational
properties emerge from the existence of an energy function of the states of the neuron outputs, termed the computational energy. For a given set of inputs, the output state of the network converges towards one of the minima of the computational energy, and thus can be used to solve optimisation problems, providing that appropriate connections between the neurons, and the external input to the neurons, can be found.

The binary Hopfield model uses two-state threshold neurons that follow a stochastic algorithm. Each neuron \( i \) has two states, characterised by the output \( V_i \) of the neuron having the value of 0 or 1. The input to each neuron comes from two sources, external inputs \( I_i \) and inputs from other neurons. The total input to neuron \( i \), \( H_i \), is given by

\[
H_i = I_i + \sum_{j=1}^{N} T_{ij} V_j
\]  

(3)

The element \( T_{ij} \) is the synaptic interconnection strength to neuron \( i \) from neuron \( j \). Hopfield has shown that, if the matrix of interconnection strengths, \( T \), is symmetric and has zero-valued diagonal elements (i.e. \( T_{ij} = T_{ji} \) and \( T_{ii} = 0 \)), then the function of the neuron output states \( E \) decreases by any state change produced by the asynchronous threshold rule \([4]\). This function, known as the computational energy function, is defined as

\[
E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} V_i T_{ij} V_j - \sum_{i=1}^{N} I_i V_i
\]  

(4)

The network updates its states according to the threshold rule until the stable state is reached, i.e. when further iterations do not change the \( V_i \) output values. This stable state corresponds to a minimum of the computational energy.

**HOPFIELD NETWORK BTC**

Recent work by the authors has developed a Hopfield neural network BTC technique \([7]\). Instead of using the block mean as a threshold to classify pixels in the block into two classes, this technique employs the computational power of the Hopfield neural network to find the bit map distribution. The technique has been shown to improve performance, in the mean square error sense, over established block truncation coding techniques.

The image to be compressed is divided into \( n \times n \) blocks, and a Hopfield network with \( N = n \times n \) neurons corresponding to the block is constructed, each two-state neuron in the network being associated with one of the pixels in the block.

Let \( V_i \) be the state of the neuron \( i \) in the network corresponding to pixel \( x(i) \), \( i = 1, 2, \ldots, N \). The value of \( V_i \) is either 0 or 1. The aim is to partition the \( N \) pixels in the block into two classes and quantise them into a two-level output in such a way that as little as possible error is introduced. An appropriate approach to this problem is to group pixels with similar intensities into the same class. An energy function \( E \) describes the network such that the lowest energy state corresponds to an effective classification of the pixels. To achieve this we define the computational energy function as

\[
E = \sum_{i=1}^{N} \sum_{j=1}^{N} V_i V_j |x(i) - x(j)|^2 + \sum_{i=1}^{N} \sum_{j=1}^{N} (1 - V_i)(1 - V_j) |x(i) - x(j)|^2
\]  

(5)

From equation (5) we can see that if two pixels are assigned to the same class, the difference between the pixel values must be small, otherwise the network will not be stabilised. Thus the network tends to encourage pixels with similar intensity values to group into the same class. Conversely, the contribution to the energy function of an arrangement whereby two pixels are assigned to different classes is zero (note that the energy function is always positive), and this means that the network has a strong tendency to place these pixels into different classes. A Hopfield network governed by the computational energy function of equation (5) is able to classify pixels in a small block into two classes effectively.

The images are divided into \( n \times n \) blocks for coding, and a Hopfield network with \( n \times n \) neurons is used for each block in turn. The synaptic interconnection strengths and external input to the neurons are given by

\[
T_{ij} = -4|x(i) - x(j)|^2
\]  

(6)

\[
I_i = 2 \sum_{j=1}^{N} |x(i) - x(j)|^2
\]  

(7)

The original pixel intensities, normalised with respect to the maximum block pixel value, are used as the initial output states of the network. The network is iterated until the stable state is reached, i.e. further iterations do not change the \( V_i \) values. The quantiser output levels are
the mean of each of the two classes, calculated according to

$$\bar{X}_0 = \frac{1}{N} \sum_{i=1}^{N} (1 - V_i) x(i)$$

$$\bar{X}_1 = \frac{1}{q} \sum_{i=1}^{N} V_i x(i)$$

where $q$ is the number of pixels in class $V_i = 1$.

When the network has reached the stable state, the value of the computational energy, $E_{sp}$, may be evaluated using

$$E_{sp} = \sum_{i=1}^{N} \sum_{j=1}^{N} |x(i) - x(j)|^2 \left| \frac{V_i V_j}{V_i + V_j} \right|$$

The value of $E_{sp}$ may be interpreted as a measure of the 'roughness' of the original block and used to determine how the block is coded, as discussed in the following section.

**HOPFIELD NETWORK BASED VARIABLE BIT RATE BTC**

In some parts of natural images, the pixel variance inside a small block, such as a $4 \times 4$ block, is very small. The bit map of such a block can be discarded and each pixel of the block may be replaced by the block mean without significantly affecting the quality of the reconstructed image. In this section we present a Hopfield network based variable bit rate BTC scheme which incorporates this feature. The value of the computational energy in the stable state of the network, $E_{sp}$, computed by the Hopfield network, is used as an indication of the image detail properties of the block. A high value of $E_{sp}$ indicates that the pixel variance inside the block is high, and in this case the ordinary Hopfield network BTC scheme is used to code the block, as described in the previous section. Conversely, a low value of $E_{sp}$ indicates that the pixel variance inside the block is low. In this case, the bit map of the block is not used - instead each pixel of the block is replaced by the block mean value in the reconstructed image. It was found in simulations that a large proportion of blocks produce a very small final energy value, and hence many blocks may be coded as low detail blocks.

The variable bit rate BTC scheme works as follows: The images are divided into $4 \times 4$ small blocks, and a Hopfield neural network as described above is used to code each block in turn. After the network has converged, the stable state energy value $E_{ss}$ is evaluated according to equation (9). If this value is greater than a preset value, the block is seen to be a high detail block, and hence a bit map and two class means are transmitted. However, if $E_{ss}$ is less than the preset threshold, the block is a low detail block, and only the average value of the pixels in the block is transmitted. To distinguish between the high and low detail blocks, a one-bit flag is needed for each $4 \times 4$ block. It can be seen that the compression ratio $C_r$ of this scheme can be evaluated as

$$C_r = \frac{8M^2}{33B_{hd} + 9B_{ld}}$$

where $B_{hd}$ is the number of high detail blocks and $B_{ld}$ is the number of low detail blocks. Here it is assumed that the image size is $M \times M$ pixels, the original image intensity is quantised using 8 bits, and each of the two means is also quantised using 8 bits.

**POSTPROCESSING**

In using the Hopfield network variable bit rate BTC scheme, the block effect may appear, i.e. the boundaries between the blocks may become visible. At a low bit rate, the block effect is a common problem for block based image coding techniques such as vector quantization and the discrete cosine transform. In our Hopfield network variable bit rate BTC technique, the high and low detail blocks are coded separately, and hence a simple smoothing technique can be applied to the boundary pixels of the low detail blocks, without significantly degrading the detail (edges) in the image.

To implement this postprocessing, we identify the image blocks which are coded as low detail blocks and apply a simple averaging filter to smooth the block boundaries. Figure 1 shows the smoothing pixels for the block side and corner pixels. Such a smoothing process will not degrade image detail, since the high detail blocks are coded separately and are not subjected to this smoothing.

![Figure 1 Smoothing pixels for averaging filter](image-url)
EXPERIMENTAL RESULTS

To demonstrate the comparative performance of the new neural network based BTC scheme, reconstructed images and coding results are presented in Figure 2. The block size used was 4 x 4 pixels and the threshold applied to $E_{th}$ was set to 25.

Compared with previous BTC methods [1,7], it is seen that the new variable bit rate BTC scheme achieves an improved compression ratio, namely an increase from 4:1 to 7.35:1. Also compared with the Hopfield BTC method [7], the improved compression ratio is achieved at the expense of a slight deterioration in MSE value, but it is seen that the difference in visual quality for the two reconstructed images (c) and (d) is virtually imperceptible.

CONCLUDING REMARKS

In this paper, we have introduced a new Hopfield neural network based BTC technique for image compression which differentiates between high detail and low detail sub-image blocks. The experimental results demonstrate the effectiveness of the new scheme. The new technique is a highly parallel structure and is therefore computationally efficient.

REFERENCES

6 Hopfield, J. J. and Tank, D.W., 'Neural computation of decisions in optimization problems', Biol. Cybern., 1985, 52, pp. 141-152
Figure 2 Experimental Results:
(a) Original image (Lena)
(b) Reconstructed image using original BTC method [1], MSE = 50.86, C_r = 4
(c) Reconstructed image using Hopfield BTC method [7], MSE = 25.05, C_r = 4
(d) Reconstructed image using new variable bit rate BTC method, MSE = 29.30, C_r = 7.35