Video Compression Using Spiht and Neural Network

[1,]Sangeeta Mishra,[2,]Sudhir Sawarkar

Abstract

Apart from the existing technology on image compression represented by series of JPEG, MPEG and H.26x standards, new technology such as neural networks and genetic algorithms are being developed to explore the future of image coding. Successful applications of neural networks to basic propagation algorithm have now become well established and other aspects of neural network involvement in this technology. In this paper different algorithms were implemented like gradient descent back propagation, gradient descent with momentum back propagation, gradient descent with adaptive learning back propagation, gradient descent with momentum and adaptive learning back propagation and Levenberg-Marquardt algorithm. The compression ratio obtained is 1.1737089:1. It was observed that the size remains same after compression but the difference is in the clarity.

Keyword: Macroblock, Neural Network,SPIHT

1. Introduction

Reducing the transmission bit-rate while concomitantly retaining image quality is the most daunting challenge to overcome in the area of very low bit-rate video coding, e.g., H.26X standards [3].[6]. The MPEG-4 [2] video standard introduced the concept of content-based coding, by dividing video frames into separate segments comprising a background and one or more moving objects. This idea has been exploited in several low bit-rate macroblock-based video coding algorithms [1][8] using a simplified segmentation process which avoids handling arbitrary shaped objects, and therefore can employ popular macroblock-based motion estimation techniques. Such algorithms focus on moving regions through the use of regular pattern templates, from a pattern codebook (Figure 1), of non-overlapping rectangular blocks of 16×16 pixels, called macroblocks (MB).

2. SPIHT

When the decomposition image is obtained, we try to find a way how to code the wavelet coefficients into an efficient result, taking redundancy and storage space into consideration. SPIHT [2] is one of the most advanced schemes available, even outperforming the state-of-the-art JPEG 2000 in some situations. The basic principle is the same; a progressive coding is applied, processing the image respectively to a lowering threshold. The difference is in the concept of zerotrees (spatial orientation trees in SPIHT). This is an idea that takes bounds between coefficients across subbands in different levels into consideration. The first idea is always the same: if there is an coefficient in the highest level of transform in a particular subband considered insignificant against a particular threshold, it is very probable that its descendants in lower levels will be insignificant too, so we can code quite a large group of coefficients with one symbol. SPIHT makes use of three lists – the List of Significant Pixels (LSP), List of Insignificant Pixels (LIP) and List of Insignificant Sets (LIS). These are coefficient location lists that contain their coordinates. After the initialization, the algorithm takes two stages for each level of threshold – the sorting pass (in which lists are organized) and the refinement pass (which does the actual progressive coding transmission). The result is in the form of a bitstream. Detailed scheme of the algorithm is presented in Fig. 3.The algorithm has several advantages. The first one is an intensive progressive capability – we can interrupt the decoding (or coding) at any time and a result of maximum possible detail can be reconstructed with one-bit precision. This is very desirable when transmitting files over the internet, since users with slower connection speeds can download only a small part of the file, obtaining much more usable result when compared to other codec such as progressive JPEG. Second advantage is a very compact output bitstream with large bit variability – no additional entropy coding or scrambling has to be applied. It is also possible to insert a watermarking scheme into the SPIHT coding domain [3] and this watermarking technique is considered to be very strong regarding to watermark invisibility and attack resiliency. But we also have to consider disadvantages. SPIHT is very vulnerable to bit corruption, as a single bit error can introduce significant image distortion depending of its location. Much worse property is the need of precise bit synchronization, because a leak in bit transmission can lead to complete misinterpretation from the side of the decoder. For SPIHT to be employed in real-time applications, error handling and synchronization methods must be introduced in order to make the codec more resilient.

3. Principle Of SPIHT Coder:

SPIHT is based on three principles: i) Exploitation of the hierarchical structure of the wavelet transform, by using a tree-based organization of the coefficients;

ii) Partial ordering of the transformed coefficients by magnitude, with the ordering data not explicitly transmitted but recalculated by the decoder; and

iii) Ordered bit plane transmission of refinement bits for the coefficient values.

This leads to a compressed bitstream in which the most important coefficients are transmitted first, the values of all coefficients are progressively refined, and the relationship between coefficients representing the same location at different scales is fully exploited for compression efficiency. SPIHT is a very effective, popular and computationally simple technique for image compression, it belongs to the next generation of wavelet encoders. SPIHT exploits the properties of the wavelet-transformed images to increase its efficiency. It offers better performance than previously reported image compression techniques. In this technique the pixels are divided into sets and subsets. Under the same assumption of zerotrees used in EZW, SPIHT coder scans wavelet coefficients along quad tree instead of subband and SPIHT identifies significance of wavelet coefficients only by the magnitude of coefficients and encodes the sign separately into a new bit stream. Initial threshold is calculated using the equation. The significance test is applied to these pixels. If these pixels are found to be significant with respect to threshold, they are taken into consideration for transmission otherwise ignored.

The Coding / decoding algorithm has following steps:

1) **Initialization:** Set n = |log2 (max (i,j){|c(i,j)|}) & transmit n.
Set the LSP as an empty set list, and add the coordinates
(i, j) ∈ □ H to the LIP and only those with descendants also to the LIS, as the type A entries.

2) **Sorting pass:**
2.1 for each entry (i, j) in the LIP do:
2.1.1 output Sn(i, j);
2.1.2 if Sn(i,j)=1, move(i,j) to the LSP and output the sign of ci,j;
2.2 for each entry (i, j) in the LIS do:
2.2.1 if the entry is of type A then
2.2.1.1 output Sn(D(i, j));
2.2.1.2 if Sn(D(i, j)) =1 then
2.2.1.2.1 for each (k, l) ε □ O(i, j) do:
2.2.1.2.1.1 output Sn(k, l);
2.2.1.2.1.2 if Sn(k, l) = 1, add (k, l) to the LSP and output the sign of ck, l;
2.2.1.2.1.3 if Sn(k, l) =0, add (k, l) to the LIP;
2.2.1.2.2 if £(i, J)≠□ 0, move (i, j) to the end of the LIS as an entry of type B and go to Step 2.2.2 ;
otherwise, remove entry (i, j) from the LIS;
2.2.2 if the entry is of type B then
2.2.2.1 output Sn(£(i, j));
2.2.2.2 if Sn(£(i, j)) = 1 then
2.2.2.2.1 add each (k, l) ε O(i, j) to the end of the LIS as entry of type A;
2.2.2.3 remove (i, j) from the LIS;

3) **Refinement pass:** for each entry (i, j) in the LSP, except those included in the last sorting pass (i.e. with same n), output the nth most significant bit of |ci, j|;

4) **Quantization step update:** decrement n by 1 and go to Step 2 if needed.

4. Neural network

1. Neural network outline

Neural network (NN) is a nonlinearity complex network system which consists of large quantities of processing unit analogizing neural cells, has the capability of approaching the nonlinear function, very strong fault-tolerant ability and the quick study speed of local network. Among them, feed forward network consists of three layers: the input layer, the hidden layer and the output layer. The hidden layer may have many layers, each layer neurons only accepts the previous layer neurons' output. And then ,this realizes the input and output nonlinear mapping relationship by adjusting the connected weighted value and the network structure[1].At present the neural network research method has formed many schools, the most wealthiest research work includes: the multi-layer network BP algorithm, Hopfield neural network model, adaptive resonance theory, self-organizing feature map theory and so on. This paper is based on researching BP neural network training

quantification parameter [2],presents a network that is better than BP (Back Propagation) network in its property of optimal approximation.

2. Feed Forward Network

A single-layer network of neurons having *R* inputs is shown below in fig. 1, full detail on the left and with a layer diagram on the right.

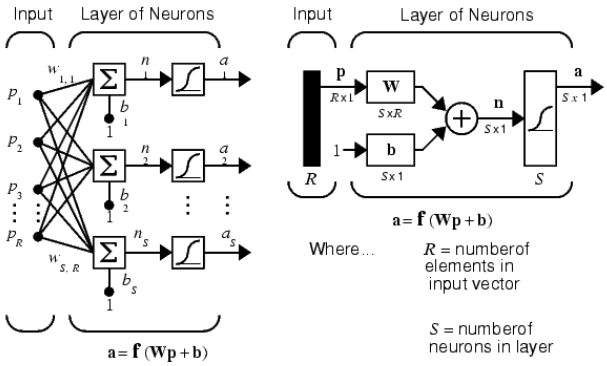

Fig.1. Feed Forward Network

Feed forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1.For multiple-layer networks we use the number of the layers to determine the superscript on the weight matrices. The appropriate notation is used in the two-layer tansig/purelin network shown in fig 2.

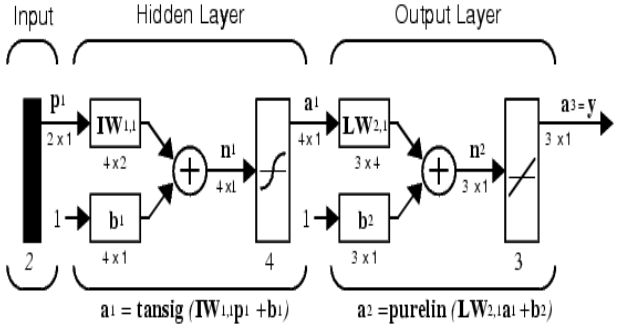

Fig.2 Multiple Layer Input

This network can be used as a general function approximation. It can approximate any function with a finite number of discontinuities, arbitrarily well, given sufficient neurons in the hidden layer.

3. Back propagation algorithm:

Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Standard back propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term *back propagation* refers to the manner in which the gradient is computed for nonlinear multilayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. There are generally four steps in the training process:

1. Assemble the training data
2. Create the network object
3. Train the network
4. Simulate the network response to new inputs

The generalized delta rule, also known as back propagation algorithm is explained here briefly for feed forward Neural Network (NN). The NN explained here contains three layers. These are input, hidden, and output Layers. During the training phase, the training data is fed into to the input layer. The data is propagated to the hidden layer and then to the output layer. This is called the forward pass of the back propagation algorithm. In forward pass, each node in hidden layer gets input from all the nodes from input layer, which are multiplied with appropriate weights and then summed. The output of the hidden node is the non- linear transformation of this resulting sum. Similarly each node in output layer gets input from all the nodes from hidden layer, which are multiplied with appropriate weights and then summed. The output of this node is the non-linear transformation of the resulting sum. The output values of the output layer are compared with the target output values. The target output values are those that we attempt to teach our network. The error between actual output values and target output values is calculated and propagated back towards hidden layer. This is called the backward pass of the back propagation algorithm. The error is used to update the connection strengths between nodes, i.e. weight matrices between input-hidden layers and hidden-output layers are updated.

4. Results And Discussions

During the testing phase no learning takes place i.e., weight matrices are not changed. Each test vector is fed into the input layer. The feed forward of the testing data is similar to the feed forward of the training data. Various algorithms were tried and tested.

Fig. 3 Original Residue Frame

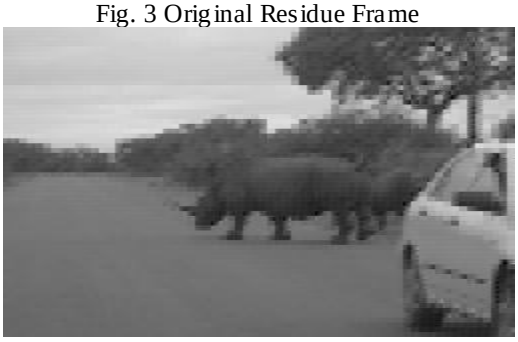

Fig.4 Frame after SPIHT with gradient descent with momentum and adaptive learning back propagation

Fig.5 Frame after SPIHT with gradient descent with adaptive learning back propagation

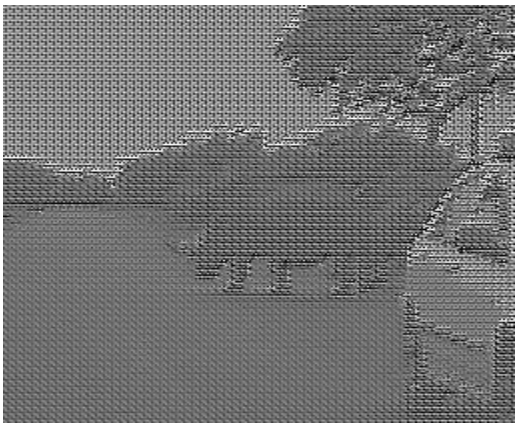

Fig.6 Frame after SPIHT with gradient descent with momentum back propagation

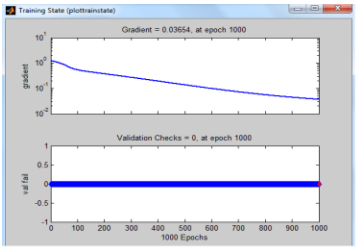

Fig.7 Traning Achieved

Fig.8 Frame after SPIHT with gradient descent back propagation

Fig.9 Frame after SPIHT with Levenberg-Marquardt algorithm

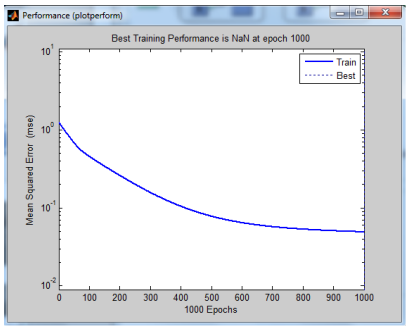

Fig.10 Performance Achieved

5. References

1. M.Vetterli and J Kovaccevic, Wavelets & Subband Coding, Prentice Hall PTR, Englewood cliffs, NJ, 1995.
2. C. Sidney Burrus, Ramesh A Gopinath, and Haitao Guo, Introduction to Wavelets & Wavelets Transforms, Prentice Hall Inc, Upper Saddle River, NJ,1998.
3. O Rioul and M Vetterli,"Wavelets and signal processing," IEEE Signal Processing Magazine, vol. 8, no. 4, pp. 14–38, 1991.
4. A. Said, W. A. Pearlman, "SPIHT Image Compression: Properties of the Method", Html
5. A. Said, W.A. Pearlman: "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6,1996.
6. J.A.Freeman and D.M.Skapura, Neural Networks:
   algorithms, Applications and Programming Techniques
   Addition-Wesley), 1991.
7. H.Demuth and M. Beale. Neural Network TOOLBOX User's Guide. For use with MATLAB. The Math Works lne.. (1998)
8. Kiamal Z. Pekmestzi. Multiplexer-Based Array Multipliers. IEEE Transcation on computers, vol, 48 , JAN (1998)
9. Hennessy. J.L and Patterson. D.A. Compter Architecture: A quantitative Approach. Morgan Kaufmanns, (1990)
10. J. Jiang. Image compression with neural networks. Signal Processing: Image Communication 14 (1999) 737-760