



G00002

A NEURAL NETWORK BASED METHOD FOR EXPONENT CODING
IN A TRANSFORM CODER FOR HIGH QUALITY AUDIO

Technical Field

5

This invention is applicable in the field of audio coders which employ exponent coding techniques to provide high levels of digital data compression.

Background Art

10

In order to more efficiently broadcast or record audio signals, the amount of information required to represent the audio signals may be reduced. In the case of digital audio signals, the amount of digital information needed to accurately reproduce the original pulse code modulation (PCM) samples may be reduced by applying a digital compression
15 algorithm, resulting in a digitally compressed representation of the original signal. The goal of the digital compression algorithm is to produce a digital representation of an audio signal which, when decoded and reproduced, sounds the same as the original signal, while using a minimum of digital information for the compressed or encoded representation.

20 Recent advances in audio coding technology have led to high compression ratios while keeping audible degradation in the compressed signal to a minimum. These coders are intended for a variety of applications, including 5.1 channel film soundtracks, HDTV, laser discs and multimedia. Description of one applicable method can be found in the Advanced Television Systems Committee (ATSC) Standard document entitled "Digital
25 Audio Compression (AC-3) Standard", Document A/52, 20 December, 1995.

In the basic approach, at the encoder the time domain audio signal is first converted to the frequency domain using a bank of filters. The frequency domain coefficients, thus generated, are converted to fixed point representation. In fixed point syntax, each
30 coefficient is represented as a mantissa and an exponent. The bulk of the compressed



- 2 -

bitstream transmitted to the decoder comprises these exponents and mantissas.

The exponents are usually transmitted in their original form. However, each mantissa must be truncated to a fixed or variable number of decimal places. The number of bits to be used for coding each mantissa is obtained from a bit allocation algorithm which may be based on the masking property of the human auditory system. Lower numbers of bits result in higher compression ratios because less space is required to transmit the coefficients. However, this may cause high quantization errors, leading to audible distortion. A good distribution of available bits to each mantissa forms the core of the advanced audio coders.

Further compression is possible by employing differential coding for the exponents. In this case, the exponents for a channel are differentially coded across the frequency range. That is, instead of sending actual values, for many exponents only the difference between adjacent exponent values is sent. Furthermore, instead of coding every exponent a single exponent value is sent for every two exponents. In the extreme case, when exponent sets of several consecutive blocks in a frame are almost identical the exponent set for the first block is only sent, and the remaining blocks in the frame reuse the previously sent values.

These different methods amount to different exponent coding schemes presently in use, and are commonly referred to as exponent strategies.

The differential exponent coding schemes presently in use limit the difference between consecutive exponents to a fixed value which is predefined by the standard. This does not take into consideration the signal characteristic. As a result, sometimes for a fast varying audio spectrum the limit is too low, which results in information loss due to truncation. At other times the limiting value may be too large for a slow varying spectrum, thereby resulting in bit wastage.

A good set of exponent strategies is therefore desirable, together with an efficient

- 3 -

processing engine which determines which exponent strategy is most suited for a particular set of exponents. The processing engine should not be too computationally intensive to allow encoders to operate in real time and run on systems having relatively small computational resources.

5

Summary of the Invention

In accordance with the present invention, there is provided a method for coding audio data comprising a sequence of exponent sets each comprising a plurality of exponents,

10 comprising the steps of:

determining a first variation of exponent values between a first exponent set in the sequence and each subsequent exponent set in said sequence;

selecting an exponent coding strategy for said first exponent set from a plurality of exponent coding strategies on the basis of said first variation; and

15 coding said first exponent set according to the selected exponent coding strategy.

Preferably each of the plurality of exponent coding strategies corresponds to a different differential coding limit.

20 In one form of the invention, the method includes selecting one of said subsequent exponent sets on the basis of said first variation and assigning an exponent re-use coding strategy to the selected exponent set and any exponent sets in said sequence between the first exponent set and the selected exponent set.

25 Preferably the method includes a step of determining a second variation between consecutive exponents in said first exponent set, wherein the exponent coding strategy for said first exponent set is selected on the basis of said first and second variations.

The step of selecting the exponent coding strategy for the first exponent set is preferably
30 performed using neural network processing. In that case, the neural network processing

- 4 -

may include a feature extraction stage in which said sequence of exponent sets is processed to determine said first variation values, a weighted routing stage in which said first variation values are weighted according to predetermined weighting values and routed to inputs of a first neural layer, a selection stage in which an output of the first neural layer is selected, and an output processing stage in which a coding strategy is assigned to said first exponent set based on the output of said selection stage and said second variation.

Preferably the feature extraction stage comprises determining the first variation values according to

$$Adiff(E_i, E_j) = (\sum_m |e_{i,m} - e_{j,m}|)/n$$

where $Adiff$ is said first variation,

E_i is said first exponent set and E_j is a subsequent exponent set with $j > i$,

$$E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1}),$$

$$E_j = (e_{j,0}, e_{j,1}, e_{j,2}, \dots, e_{j,n-1}),$$

n is an integer representing the number of exponents in a said set of exponents,

$$m = 0, 1, 2, \dots, n-1.$$

Preferably the processing carried out by the weighted routing stage and first neural layer includes determining

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{b-1} \end{bmatrix} = \Gamma \left(\begin{bmatrix} 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & w_{1,1} & 0 & \dots & \dots & 0 \\ 0 & w_{2,1} & w_{2,2} & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & w_{b-1,1} & w_{b-1,2} & \dots & \dots & w_{b-1,b-1} \end{bmatrix} \begin{bmatrix} 0 \\ z_1 \\ z_2 \\ \dots \\ z_{b-1} \end{bmatrix} - \begin{bmatrix} 0 \\ T_1 \\ T_2 \\ \dots \\ T_{b-1} \end{bmatrix} \right)$$

the operator $\Gamma[\cdot]$ is defined as:

- 5 -

$$\Gamma \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \dots \\ \dots \\ \gamma_{b-1} \end{bmatrix} = \begin{bmatrix} f(\gamma_0) \\ f(\gamma_1) \\ f(\gamma_2) \\ \dots \\ \dots \\ f(\gamma_{b-1}) \end{bmatrix}$$

where $f(\gamma_i)$ is +1 if $\gamma_i \geq 0$ else it is 0,

Y represents outputs of the first neural layer,

T are threshold values determined during a training phase,

w are weighting values determined during the training phase, and

5 b is the number of exponent sets in the sequence comprising said data.

In one form of the invention, the selection stage comprises selecting an output y_a of the first neural layer such that $y_a = 1$ and a is maximum for $i < a < b$. The plurality of exponent coding strategies may comprise strategies $S_1, S_2, \dots, S_c$, where $c \leq b$,

10 corresponding to respective differential coding limits 1, 2, $\dots, c$, wherein the exponent coding strategy S_γ assigned to said first exponent set E_i is selected according to

$$\gamma = \max[\min(a+1, \sigma(E)), 1]$$

where $\sigma(E) = \text{floor}((\sum_j \|e_{i,j} - e_{i,j-1}\|/n) + 0.5)$.

15 The exponent sets in the sequence between $i+1$ and a , inclusive, can then be assigned the exponent re-use coding strategy.

The present invention also provides a method for processing data in an audio data encoder, the data comprising a sequence of exponent sets each comprising a plurality of exponents,

20 comprising the steps of:

determining a first variation of exponent values within a first exponent set;

determining a second variation of exponent values between said first exponent set and each subsequent exponent set in said sequence; and

- 6 -

assigning an exponent coding strategy to the first exponent set based on the determined first and second variations.

The invention also provides a digital audio encoder in which audio data is transformed into 5 coefficients having mantissas and exponents arranged in a sequence of sets, having:

a first variation processor coupled to receive the exponents of sets from said sequence and determine a first variation of exponent values between a first set and a plurality of subsequent sets in the sequence;

a second variation processor coupled to receive the exponents of said first set and 10 determine a second variation between consecutive exponent values within said first set; and

a neural network processor coupled to receive said first and second variations and select and assign an exponent coding strategy to said first set from a plurality of coding strategies on the basis of said first and second variations.

15

Brief Description of the Drawings

The invention is described in greater detail hereinafter, by way of example only, with reference to several embodiments, and having regard to the accompanying drawings,

20 wherein:

Figure 1 is a diagrammatic illustration of the data structure of an encoded audio data bit stream, having frames and blocks;

Figure 2 is a functional block diagram of an audio encoder;

Figure 3 is a diagrammatic illustration of mapping of exponent sets to coding 25 strategies; and

Figure 4 is a diagrammatic illustration of a neural network system for selection of exponent coding strategies

Detailed Description of the Preferred Embodiments

30

- 7 -

Suppose a frame comprises b blocks and a block B_i is composed of an ordered exponent set $E_i (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1})$ and mantissa set $M_i (m_{i,0}, m_{i,1}, m_{i,2}, \dots, m_{i,n-1})$. If the elements in E_i are very similar to those in $E_{i+1}, E_{i+2}, \dots, E_{i+k}$, then only exponent set E_i needs to be coded into the bitstream. Since the final question is more a matter of

5 perception (the coding must not cause any audible degradation) rather than an exact closed form mathematical equation, a stochastic approach is suitable. Neural Networks are known to solve such problems efficiently with minimum computation requirements.

The maximum difference value allowed between exponents should be based on the

10 variation in the exponents and the number of blocks over which the exponent set is to be reused. Allowing a higher value of maximum allowed difference results in less coding error. If an exponent value is to be reused several times it is necessary to code it with minimal error or else the accumulated error over several blocks could be quite appreciable.

15

Consider a set of exponent coding strategies $\{S_0, S_1, S_2, \dots\}$. Let strategy S_0 be defined as "reuse". That is, if exponent set $E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1})$ uses strategy S_0 for coding then essentially no exponent is transmitted. Instead the receiver is directed to use exponents of E_{i-1} as those of exponent E_i . Next, let S_1 be defined as an exponent coding

20 strategy where all exponents are differentially encoded with the constraint that the difference between any two consecutive exponents is always -1, +1 or 0. That is, for exponent coding strategy S_1 , the maximum allowed difference, L , equals ± 1 . Similarly, for strategy S_2 , let the maximum allowed difference, L , be ± 2 , and so on for $S_3, S_4, \dots$

25

The inputs $(E_0, E_1, E_2, \dots, E_{b-1})$ are presented to the neural network system. The output $(o_0, o_1, o_2, \dots, o_{b-1})$ of the system is the exponent strategy corresponding to each exponent set.

- 8 -

The input to an audio coder comprises a stream of digitised samples of the time domain analog signal, which is illustrated diagrammatically in Figure 1. The bit stream comprises a sequence of frames, with each frame consisting of b blocks identified as $(B_0, B_1, B_2, \dots, B_{b-1})$. Each block contains n consecutive samples of data. A frame is an independent
 5 entry, such that there is no inter-frame data sharing. This facilitates splicing of encoded data at the frame level, and rapid recovery from transmission error. By grouping blocks to form a frame coding efficiency is increased since redundant data is shared, as permitted by signal conditions, across the frame. The block size in a particular coding standard is usually determined based on spectral and temporal resolution requirements for the audio
 10 signal.

Figure 2 illustrates a general block diagram of an audio encoder 50. Within any block B_i the n samples form an ordered set $(s_{i,0}, s_{i,1}, s_{i,2}, \dots, s_{i,n-1})$. These time domain samples $(s_{i,0}, s_{i,1}, s_{i,2}, \dots, s_{i,n-1})$ are converted to the frequency domain using an analysis filter bank
 15 52. The frequency domain coefficients, thus generated, form another ordered set which can be identified as $(c_{i,0}, c_{i,1}, c_{i,2}, \dots, c_{i,n-1})$. Here $c_{i,0}$ is the lowest frequency (dc) component while $c_{i,n-1}$ is the highest frequency component of the signal.

Audio compression essentially entails finding how much of the information in the set $(c_{i,0}, c_{i,1}, c_{i,2}, \dots, c_{i,n-1})$ is necessary to reproduce the original analog signal at the decoder with
 20 minimal audible distortion.

The coefficient set $(c_{i,0}, c_{i,1}, c_{i,2}, \dots, c_{i,n-1})$ is next converted into floating point format, where each coefficient $c_{i,j}$ is represented by an exponent $e_{i,j}$ and mantissa $m_{i,j}$. By
 25 definition, $e_{i,j} = \text{floor}(\log_2(c_{i,j}))$. Also, mantissa $m_{i,j} = (c_{i,j})/2^{e_{i,j}}$. The exponents and mantissas form two ordered sets $E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1})$ and $M_i = (m_{i,0}, m_{i,1}, m_{i,2}, \dots, m_{i,n-1})$. Therefore, each coefficient $c_{i,j}$ is related to the exponent $e_{i,j}$ and mantissa $m_{i,j}$ by the relation $c_{i,j} = m_{i,j} * 2^{e_{i,j}}$.

The exponent set is usually transmitted in its original form. However, the mantissa is truncated to a fixed or variable number of decimal places. Suppose mantissa $m_{i,j}$ is transmitted as $b_{i,j}$ number of bits. The value $b_{i,j}$ is usually obtained from a bit allocation algorithm which for advanced psychoacoustic coders may be based on the masking property of the human auditory system. Low value of $b_{i,j}$ results in higher compression ratio because less space is required to transmit the coefficients. However, this causes very high quantization error leading to audible distortion. A good distribution of available bits to each mantissa forms the core of the most advanced encoders.

Further compression is made possible in audio coders by application of certain coding techniques for exponents. Since most music is characterised by slowly changing spectrum across audio blocks, it is possible to reuse the exponents of a block B_i as exponents of the next few following blocks within the frame. That is, if the elements in exponent set E_i are very similar to the corresponding elements in sets E_{i+1} , E_{i+2} , ..., E_{i+n} , then only exponent set E_i needs to be coded into the bitstream. A bit can be set to indicate to the decoder to reuse E_i for the next n blocks in the frame. The similarity of set E_i to another set E_j is, in general, a difficult problem to solve.

In addition to reuse, the exponents for a channel are usually differentially coded across the frequency range. That is, instead of sending the actual values, only the difference between the exponents are sent. However, as mentioned, the exponent coding schemes limit the allowable difference between adjacent exponents in the sequence to a fixed value, called L , which is predefined by the standard. This does not take into consideration the actual signal characteristics. As a result, sometimes for fast varying spectrum the limit is too low which results in information loss due to truncation. At other times the limiting value may be too large for a slow varying spectrum, thereby resulting in bit wastage.

Instead of allowing variation in value of L , some existing coders modify the frequency resolution in order to provide varying levels of compression for exponents. For example,

- 10 -

instead of sending delta values for every element in set $(e_{i,0}, e_{i,1}, e_{i,2}, e_{i,3}, e_{i,4}, \dots, e_{i,n-1})$, only delta for every 2 (or 3 or 4 etc.) frequency coefficients are sent. That is, the transmitted set is $(e_{i,0}, e_{i,2}, e_{i,4}, \dots, e_{i,n-1})$. At the receiver the values for $e_{i,1}, e_{i,3}, e_{i,5}$ are obtained by interpolation of the received values. However, this scheme is essentially a
 5 subset of the former scheme of varying the value of L.

To pose the problem more formally, suppose a set of exponent coding strategies is defined as $\{S_0, S_1, S_2, \dots\}$. Let strategy S_0 , be defined as "reuse". That is if exponent set E_i uses strategy S_0 for coding then essentially no exponent is transmitted but instead the
 10 receiver is directed to use exponents of E_{i-1} as those of exponent E_i . Next, let S_1 be the exponent coding strategy where all exponents are differentially encoded with the constraint that the difference between any two consecutive exponents is always -1, +1 or 0. That is, for exponent coding strategy S_1 , the maximum allowed difference, L, equals +/-1. Similarly, for strategy S_2 , let the maximum allowed difference, L, be +/-2, and so on for
 15 $S_3, S_4, \dots$.

Based on the above discussion, the problem of exponent strategy assignment can be formerly stated as a injective mapping of each exponent set E_i within a frame to a suitable strategy S_j such that maximum compression is achieved with minimal audible distortion.
 20 A mapping of exponent sets to coding strategies is illustrated diagrammatically in Figure 3. Since each frame is an independent entry, exponent set E_0 is not allowed to use strategy S_0 or else it would attempt to use the exponent information in the previous frame.

For example, suppose a frame consists of six blocks with exponent sets E_0, E_1, E_2, E_3, E_4
 25 and E_5 , and the assigned strategy is S_3, S_0, S_0, S_0, S_0 and S_1 , respectively. This means exponents in set E_0 are differentially encoded with the constraint that the maximum allowed difference is +/-5 (by previous definition of strategy S_3). For blocks 1, 2, 3 and 4, no exponent value is sent as they are instructed to reuse the exponents from the previously transmitted set. Block B_4 's exponents, that is E_4 , are transmitted as

- 11 -

differentially encoded values with the constraint that the maximum allowed difference is ± 1 (definition of strategy S_i).

The optimal mapping is virtually impossible to find due to the complex behaviour of the system, large numbers of competing parameters, and the fact that optimality depends on perceived quality of reproduced music rather than on a closed form equation with a minimal value. Moreover, there is a long-standing debate on trade-offs between quality of final music, compression ratio achieved and computational requirements of the system. Therefore, having formally outlined the problem space, a stochastic approach is used to provide the solution. A Neural Network based solution is described herein.

Neural Network System

A neural network's ability to perform computation is based on the effort to reproduce the flexibility and power of the human brain by artificial means. Neural computation is performed by a dense mesh of computing nodes and connections. They operate collectively and simultaneously on most or all data inputs. The basic processing elements of the neural network are called nodes, or simply neurons. They can be considered as threshold units that fire when their total weighted input exceeds certain bias levels. Neurons are often organised in several layers, with neurons in consecutive layers often being connected together. Each connection strength is expressed by a numerical value called *weight*, the value of which is determined by the specific system which the network attempts to model.

Figure 4 is a diagrammatic illustration of a neural network system for selection of exponent coding strategies. The inputs to the neural network system are the exponent sets $(E_0, E_1, E_2, \dots, E_{b-1})$, where $E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1})$. The output $(o_0, o_1, o_2, \dots, o_{b-1})$ is the exponent strategy corresponding to each exponent set. Starting from E_n the strategy for each exponent set is determined. Suppose the exponent strategy for E_i is to be

determined. The exponent coding strategy for E_i depends on:

1. The number of times E_i will be reused across the blocks in the frame. If an exponent value is to be reused several times it is necessary to code it with minimal error or else the accumulated error over several blocks could be quite appreciable.
2. The variation between consecutive elements in the set E_i . For a spectrum with large variations across the frequency range the limit L must be high to minimise information loss due to truncation. For a spectrum with low variation a low value of L is adequate and this results in better compression ratio.

A two stage network is used to take both these aspects into consideration when deciding on the strategy for an exponent set. The first criteria is of the number of blocks over which an exponent set is reused. This is dealt with primarily by the first layer. If the elements in E_i are very similar to those in E_{i+1} , E_{i+2} , ..., E_{i+a} then only exponent set E_i needs to be coded into the bitstream. A flag can be set to indicate to the decoder to *reuse* E_i for the next a blocks in the frame.

Feature Extraction

To find the number of blocks over which the same exponent set can be safely used without audible distortion in signal, a measuring function for similarity is required. The correlation coefficient:

$$r(E_i, E_j) = \frac{\sum_m (e_{i,m} - \bar{e}_i) * (e_{j,m} - \bar{e}_j)}{\sqrt{\sum_m (e_{i,m} - \bar{e}_i)^2 * (e_{j,m} - \bar{e}_j)^2}} \quad \text{Eq. 1}$$

is mathematically robust method for computing similarity. But it is too computationally

intensive for a practical system. Let $Adiff(E_i, E_j)$ be a function that determines the similarity between the elements of the set E_i and E_j and returns a value which shows the degree of difference. The similarity function $Adiff$ (mean absolute difference) is defined as:

$$Adiff(E_i, E_j) = (\sum_m |e_{i,m} - e_{j,m}|)/n \quad \text{Eq. 2}$$

The values $Adiff(E_i, E_{i+1}), Adiff(E_i, E_{i+2}), \dots, Adiff(E_i, E_{b-1})$ form the input $z_1, z_2, \dots, z_{b-1}$, respectively, to the first stage of neurons. Input z_0 is always set to 0, as discussed hereinbelow. Since this step extracts only that characteristic of the input which is instrumental in the decision process of the network, it is termed as feature-extraction.

Reuse Strategy

The first stage of the network determines the maximum number of exponent sets that are considerably similar to a given set E_i . The topology of the network and connection weights are such that the neuron y_1 fires (corresponding output is 1) if z_1 is within acceptable limit. Similarly, y_2 fires if both z_1 and z_2 are within acceptable limits. Since $z_1 = Adiff(E_i, E_{i+1})$, it implies that if y_1 fires then E_i and E_{i+1} are very similar. Similarly, if y_2 fires it implies E_{i+1} and E_{i+2} are both very similar to E_i . As z_0 and the threshold value for y_0 are both zero, y_0 always fires.

Each neuron in the first layer therefore computes the weighted sum of its input. If the weighted sum is less than or equal to the threshold value, the neuron fires thus making the output 1. If the threshold value is larger than the weighted sum of inputs it means that not all inputs to this neuron are similar to E_i and thus the neuron does not fire. Using matrix notation the output of the first stage of the network is:

- 14 -

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{b-1} \end{bmatrix} = \Gamma \left(\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & w_{1,1} & 0 & \dots & 0 \\ 0 & w_{2,1} & w_{2,2} & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & w_{b-1,1} & w_{b-1,2} & \dots & w_{b-1,b-1} \end{bmatrix} \begin{bmatrix} 0 \\ z_1 \\ z_2 \\ \dots \\ z_{b-1} \end{bmatrix} - \begin{bmatrix} 0 \\ T_1 \\ T_2 \\ \dots \\ T_{b-1} \end{bmatrix} \right) \quad \text{Eq. 3}$$

the operator $\Gamma[\cdot]$ is defined as:

$$\Gamma \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{b-1} \end{bmatrix} = \begin{bmatrix} f(y_0) \\ f(y_1) \\ f(y_2) \\ \dots \\ f(y_{b-1}) \end{bmatrix} \quad \text{Eq. 4}$$

where $f(y_i)$ is +1 if $y_i \geq 0$ else it is 0. The weight ($w_{i,j}$) and the threshold (T_i) values are determined from the training phase of the neural network.

- 5 Perceptron learning rule is used for training the network. In this form of supervised training, the learning signal is a function of the difference between the desired and the actual neuron's response. Before the training begins, the weights are usually initialised to small random values (small values are necessary to avoid biased initial state). An input vector is presented to the system and the output is computed. The output from the
- 10 network is compared with the desired output. The desired output was determined by extensive simulation and listening experiments. If the actual and the desired values are different the weights need to be adjusted to condition the network to imitate the desired behaviour. The weights and the threshold are adjusted according to Eq. 5 and Eq. 6, respectively. Here d_j is the desired output. The learning rate, $\alpha = 0.11$, has been

- 15 -

experimentally found to be most suited for this application.

$$\Delta w_{j,k} = \alpha [d_j - f(\sum_i (w_{ji} * z_i) - T_j)] * z_k \quad \text{Eq.5}$$

$$\Delta T_j = \alpha [d_j - f(\sum_i (w_{ji} * z_i) - T_j)] * 0.1 \quad \text{Eq.6}$$

In this system, the threshold requires a multiplication by 0.1 for maintaining same rate of adjustments for both thresholds and weights. A good set of training vectors and well adjusted desired output are crucial to the correct setting of the weights.

5

An example of weights and thresholds of a trained system for a transform coder with six audio blocks is provided below:

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .05 & 0 & 0 & 0 & 0 \\ 0 & .91 & .23 & 0 & 0 & 0 \\ 0 & 1.2 & .96 & .29 & 0 & 0 \\ 0 & .96 & .92 & .59 & .20 & 0 \\ 0 & 1.2 & 1.0 & .94 & .88 & .28 \end{bmatrix} \quad T = \begin{bmatrix} 0 \\ .02 \\ .33 \\ .71 \\ .75 \\ 1.2 \end{bmatrix}$$

Analysing the weight matrix W , an interesting behaviour becomes apparent. Along any row the weights decrease from left to right. The reason for this is as follows. Consider

10 output y_2 .

$$y_2 = f(w_{2,1} * z_1 + w_{2,2} * z_2 - T_2)$$

$$= f(w_{2,1} * \text{Adiff}(E_i, E_{i+1}) + w_{2,2} * \text{Adiff}(E_i, E_{i+2}) - T_2) \quad \text{Eq.7}$$

15

If we consider the spectrum of a signal across time, the spectrum shape at time t_{i+1} will be intermediate between spectrum shapes at time t_i and t_{i+2} , assuming $\Delta t = t_{i+1} - t_i = t_{i+2} - t_{i+1}$ is small enough. Therefore, if E_i reflects the spectral shape at time t_i and similarly

- 16 -

$E_{i,t+1}$ and $E_{i,t+2}$ the shape at $t_{i,t+1}$ and $t_{i,t+2}$, respectively, then since the spectrum is changing over time, in general $Adiff(E_i, E_{i,t+1}) \leq Adiff(E_i, E_{i,t+2})$. In Eq. 7, to level this discrepancy the multiplicative weight for $Adiff(E_i, E_{i,t+1})$ is larger than that for $Adiff(E_i, E_{i,t+2})$.

5 In effect, the similarity function for estimating similarity of two exponent sets, $Adiff(E_i, E_j)$, now has a multiplicative factor $\hat{\nu}(i,j) = w_{i,j}$. The new similarity function $\hat{\nu}(i,j) * Adiff(E_i, E_j)$, judges similarity of two sets based on not only the mean absolute difference between the elements, but also the time difference expressed as difference in block number, that exists between the two exponents sets.

10

Two exponents sets closer in time are expected to match much more than two sets with much greater time difference, in order for the similarity function to generate the same value. The fact that the network was able to automatically detect this peculiar characteristic shows its strength and our correctness in modelling of the system. It is to be
 15 noted that the system is extremely robust. In an existing system new weight vectors, obtained from perhaps superior training data and training-rule, can always be updated to provide better results.

Winner Takes all Strategy

20

To the careful reader it is evident that at any one time it is possible that more than one neuron fires. The objective is to determine the maximum number of exponent sets which can be identified as similar to given set E_i . In the winner-takes-all strategy, starting from output y_{n-1} and proceeding upward, each neuron is analysed to find the first one that has
 25 fired. Suppose y_i is the first one from the bottom to have fired. Neuron y_i is declared winner and its output is left as 1. All other neurons in this layer are now switched off. As a result the second layer's output now entirely dependent on neuron y_i .

From the network topology and the previous discussions, it is evident that y_{n-1} always fires.

- 17 -

This ensures that there is always at least one winner. Now, neuron y_1 fires if exponent sets E_i and E_{i+1} are found by the first layer to be very similar. Similarly, y_2 fires if exponent set E_{i+1} and E_{i+2} are both very similar to E_i .

5 In general, y_a fires when E_{i+a-1} , E_{i+a-2} , E_{i+a-3} ..., and E_{i+a} are all very similar to E_i .

Therefore, exponent set E_i is only coded and transmitted into the bitstream. Exponent sets E_{i+1} , E_{i+2} , E_{i+3} ..., E_{i+a} all are forced to reuse the values in E_i (that is, all are assigned strategy S_0).

10 The exact strategy for E_i still needs to be determined from the available options $\{S_1, S_2, S_3 \dots\}$. A simple technique would be to always use a fixed strategy for E_i (e.g. S_2) to provide minimal complexity.

A slight improvement on it would be to choose a strategy depending on the number of
15 blocks over which the exponent set E_i would be reused. Note that if y_a is the winner then exactly $a+1$ blocks are using the same exponent sets. If an exponent value is to be reused several times it is meaningful to code it with minimal error or else the accumulated error over several blocks could be quite appreciable. Therefore one could choose strategy S_{a+1} , where y_a is the winner neuron. Thus by this process, if the winner neuron is y_a , the
20 strategy for E_i , E_{i+1} , E_{i+2} , E_{i+3} ..., E_{i+a} is S_{a+1} , S_0 , S_0 , S_0 , ..., S_0 , respectively.

The problem with this improved version is that it does not take the variations of the elements within the exponent set into consideration. A much improved coding strategy for E_i would take into consideration not only the number of reuse blocks but also the
25 variation within E_i .

Mean absolute deviation, defined as $Adev(E_i) = (\sum_j \|e_{i,j} - \bar{e}_i\|) / n$ where $\bar{e}_i = \sum e_{i,j} / n$, could be used to provide a measure of variation. However, because finally differential coding is to be done, it is much more meaningful to base the strategy on the mean average difference

between consecutive exponents, rather than the deviation of exponents from the mean.

Consequently, we define mean absolute deviation as:

$$\sigma(E_i) = \text{floor}((\sum_j \|e_{i,j+1} - e_{i,j}\|/n) + 0.5) \quad \text{Eq. 8}$$

5

Finally, the exponent coding strategy recommended for E_i is S_γ , where γ is defined by Eq. 9 below:

$$\gamma = \max[\min(a+1, \sigma(E_i)), 1] \quad \text{Eq. 9}$$

10

By this definition, γ is always greater or equal 1. This ensures that reuse is never suggested for E_i . If variation value $\sigma(E_i)$ is greater than the number of reuse blocks, the value $a+1$ dictates which strategy is finally chosen. This provides a safety measure by preventing explosion in bit usage for large varying signals with limited reuse. When
15 variation is low, as guided by $\sigma(E_i)$, bit usage is optimised by using only as many bits for the differential limit as required and not being overwhelmed by the large number of reuse blocks.

At the end of this process the strategy for $E_1, E_{1+1}, E_{1+2}, E_{1+3}, \dots, E_{1+a}$, is $S_\gamma, S_0, S_0, S_0, \dots, S_0$, respectively. Now strategy for $E_{1+a+1}, E_{1+a+2}, E_{1+a+3}, \dots, E_{b-1}$, needs to be
20 determined. Treating E_{1+a+1} as E_1 the whole process is repeated.

The deviation values $\sum_j \|e_{i,j+1} - e_{i,j}\|/n$ for some benchmark music streams are given in the tables below. From the statistics it is evident that there is enough variation in frequency
25 domain for commonly occurring music streams to warrant a flexible limit for L in a differential coding scheme. Moreover, from the range of values, it can be concluded that absolute deviation can be effectively used to determine the maximum differential limit, and hence the exponent strategy.

- 19 -

Adev	3.5	2.7	3.7	2.8	3.7	2.3	2.3
Reuse	2	1	3	3	2	4	3

Castanets

5 Adev	1.6	2.5	1.8	2.8	2.4	1.5	1.4
Reuse	6	6	6	4	6	6	6

Pop

10 Adev	4.2	3.13	3.03	2.6	2.5	2.7	0.9
Reuse	2	1	4	4	6	3	1

Harp

Note the third and fourth column in the table for castanets. The exponents are reused across three blocks in both cases but the variations are radically different. In the first case a differential limit of 4 is necessary while in the second a limit of 3 should be adequate. This shows that improvement in quality and coding efficiency can be achieved using this technique proposed above.

In summary of the above, referring again to the audio encoder shown in Figure 2, the frequency domain coefficients which are obtained from the analysis filter bank 52 are split into their constituent mantissas and exponents, and the exponents extracted (54) and passed to an exponent strategy processor 56. The exponent strategy processor 56 embodies the neural network system described hereinabove and illustrated diagrammatically in Figure 4. The exponent strategy processor 56 receives the extracted exponents and for each operation of the neural network system determines an appropriate coding strategy for at least one of the sets of exponents received. Each subsequent operation of the exponent strategy processor 56 then begins with the next exponent set not yet allocated a strategy. The exponents and assigned strategies are passed then to an exponent coder 58 which differentially codes the exponent sets according to the assigned strategies. The coded

- 20 -

exponents are passed to a quantizer 60 where they are quantized together with the corresponding mantissas for packing into the output bitstream. Where an exponent set has been allocated a re-use strategy, quantization is not required, and accordingly a "re-use flag" is passed directly from the exponent coder 58 to the bitstream packer 62.

5

The foregoing detailed description of the invention has been presented by way of example only, and is not intended to be considered limiting to the invention as defined in the claims appended hereto.

Claims:

1. A method for processing data in an audio data encoder, the data comprising a sequence of exponent sets each comprising a plurality of exponents, comprising the steps
5 of:
 - determining a first variation of exponent values within a first exponent set;
 - determining a second variation of exponent values between said first exponent set and each subsequent exponent set in said sequence; and
 - assigning an exponent coding strategy to the first exponent set based on the
10 determined first and second variations.
2. A method as claimed in claim 1, wherein the exponent coding strategy is assigned from a plurality of exponent coding strategies having different differential coding limits.
- 15 3. A method as claimed in claim 2, including a step of coding said first exponent set according to the assigned exponent coding strategy.
4. A method as claimed in claim 3, including a step of assigning an exponent coding strategies to at least one subsequent exponent set based on the corresponding determined
20 second variation.
5. A method as claimed in claim 4, wherein the plurality of exponent coding strategies includes an exponent set re-use strategy which is assigned to the at least one subsequent exponent set.
- 25 6. A method as claimed in claim 5, including a step of coding said first exponent set and said at least one subsequent exponent set according to the corresponding assigned coding strategies

- 22 -

7. A method as claimed in claim 2 or 6, wherein the steps of determining the first and second variations are performed utilising neural network processing.

8. A method as claimed in claim 7, wherein the neural network processing includes
5 first and second neural layers.

9. A method as claimed in claim 7, wherein the neural network processing comprises a feature extraction stage in which said sequence of exponent sets is utilised to determine said second variations, a weighted routing stage in which said second variations are
10 weighted according to predetermined weighting values and routed to inputs of a first neural layer, a selection stage in which an output of the first neural layer is selected, and an output processing stage in which a coding strategy is assigned to said first exponent set based on said first variation and the output of said selection stage.

15 10. A method as claimed in claim 9, wherein a coding strategy is assigned to at least one subsequent exponent set on the basis of the output of said selection stage.

11. A method as claimed in claim 10, wherein the coding strategy assigned to the at least one subsequent exponent set is an exponent re-use strategy.

20

12. A method as claimed in claim 9, wherein the feature extraction stage comprises determining

$$\text{Adiff}(E_i, E_j) = (\sum_m |e_{i,m} - e_{j,m}|)/n$$

where Adiff is said second variation,

25 E_i is said first exponent set and E_j is a subsequent exponent set with $j > i$,

$$E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1}),$$

$$E_j = (e_{j,0}, e_{j,1}, e_{j,2}, \dots, e_{j,n-1}),$$

n is an integer representing the number of exponents in a said set of exponents,

- 23 -

$$m = 0, 1, 2, \dots, n-1.$$

13. A method as claimed in claim 12, wherein the processing carried out by the weighted routing stage and first neural layer includes determining

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ \dots \\ y_{b-1} \end{bmatrix} = \Gamma \left(\begin{bmatrix} 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & w_{1,1} & 0 & \dots & \dots & 0 \\ 0 & w_{2,1} & w_{2,2} & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & w_{b-1,1} & w_{b-1,2} & \dots & \dots & w_{b-1,b-1} \end{bmatrix} \begin{bmatrix} 0 \\ z_1 \\ z_2 \\ \dots \\ \dots \\ z_{b-1} \end{bmatrix} - \begin{bmatrix} 0 \\ T_1 \\ T_2 \\ \dots \\ \dots \\ T_{b-1} \end{bmatrix} \right)$$

5 the operator $\Gamma[\bullet]$ is defined as:

$$\Gamma \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \dots \\ \dots \\ \gamma_{b-1} \end{bmatrix} = \begin{bmatrix} f(\gamma_0) \\ f(\gamma_1) \\ f(\gamma_2) \\ \dots \\ \dots \\ f(\gamma_{b-1}) \end{bmatrix}$$

where $f(\gamma_i)$ is +1 if $\gamma_i \geq 0$ else it is 0,

Y represents outputs of the first neural layer,

T are threshold values determined during a training phase,

w are weighting values determined during the training phase, and

10 b is the number of exponent sets in the sequence comprising said data.

- 14 A method as claimed in claim 13, wherein the selection stage comprises selecting an output y_a of the first neural layer such that $y_a = 1$ and a is maximum for $i < a < b$.

- 24 -

15. A method as claimed in claim 14, wherein the plurality of exponent coding strategies comprises strategies $S_1, S_2, \dots, S_c$, where $c \leq b$, corresponding to differential coding limits 1, 2, ..., c.

5 16. A method as claimed in claim 15, wherein the exponent coding strategy S_γ assigned to said first exponent set E_i is selected according to

$$\gamma = \max[\min(a+1, \sigma(E)), 1]$$

where $\sigma(E_i) = \text{floor}((\sum_j \|e_{i,j+1} - e_{i,j}\|/n) + 0.5)$.

10 17. A method for coding audio data comprising a sequence of exponent sets each comprising a plurality of exponents, comprising the steps of:

determining a first variation of exponent values between a first exponent set in the sequence and each subsequent exponent set in said sequence;

selecting an exponent coding strategy for said first exponent set from a plurality of

15 exponent coding strategies on the basis of said first variation; and

coding said first exponent set according to the selected exponent coding strategy.

18. A method as claimed in claim 17, wherein each of the plurality of exponent coding strategies corresponds to a different differential coding limit.

20

19. A method as claimed in claim 17, including selecting one of said subsequent exponent sets on the basis of said first variation and assigning an exponent re-use coding strategy to the selected exponent set and any exponent sets in said sequence between the first exponent set and the selected exponent set.

25

20. A method as claimed in claim 17, 18 or 19, including a step of determining a second variation between consecutive exponents in said first exponent set, wherein the exponent coding strategy for said first exponent set is selected on the basis of said first and second variations.

- 25 -

21. A method as claimed in claim 17, wherein the step of selecting the exponent coding strategy for said first exponent set is performed utilising neural network processing.

5 22. A method as claimed in claim 20, wherein the step of selecting the exponent coding strategy for said first exponent set is performed utilising neural network processing.

23. A method as claimed in claim 22, wherein the neural network processing
10 comprises a feature extraction stage in which said sequence of exponent sets is processed to determine said first variation values, a weighted routing stage in which said first variation values are weighted according to predetermined weighting values and routed to inputs of a first neural layer, a selection stage in which an output of the first neural layer is selected, and an output processing stage in which a coding strategy is assigned to said
15 first exponent set based on the output of said selection stage and said second variation.

24. A method as claimed in claim 23, wherein a coding strategy is assigned to at least one subsequent exponent set on the basis of the output of said selection stage.

20 25. A method as claimed in claim 24, wherein the coding strategy assigned to the at least one subsequent exponent set is an exponent re-use strategy.

26. A method as claimed in claim 23, wherein the feature extraction stage comprises determining the first variation values according to

25
$$\text{Adiff}(E_i, E_j) = (\sum_m |e_{i,m} - e_{j,m}|)/n$$

where Adiff is said first variation,

E_i is said first exponent set and E_j is a subsequent exponent set with $i > j$,

$$E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1}),$$

$$E_j = (e_{j,0}, e_{j,1}, e_{j,2}, \dots, e_{j,n-1}),$$

- 26 -

n is an integer representing the number of exponents in a said set of exponents,

$$m = 0, 1, 2, \dots, n-1.$$

- 5 27. A method as claimed in claim 26, wherein the processing carried out by the weighted routing stage and first neural layer includes determining

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{b-1} \end{bmatrix} = \Gamma \left(\begin{bmatrix} 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & w_{1,1} & 0 & \dots & \dots & 0 \\ 0 & w_{2,1} & w_{2,2} & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & w_{b-1,1} & w_{b-1,2} & \dots & \dots & w_{b-1,b-1} \end{bmatrix} \begin{bmatrix} 0 \\ z_1 \\ z_2 \\ \dots \\ z_{b-1} \end{bmatrix} - \begin{bmatrix} 0 \\ T_1 \\ T_2 \\ \dots \\ T_{b-1} \end{bmatrix} \right)$$

the operator $\Gamma[\cdot]$ is defined as:

$$\Gamma \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{b-1} \end{bmatrix} = \begin{bmatrix} f(y_0) \\ f(y_1) \\ f(y_2) \\ \dots \\ f(y_{b-1}) \end{bmatrix}$$

where $f(y)$ is +1 if $y \geq 0$ else it is 0,

Y represents outputs of the first neural layer,

10 T are threshold values determined during a training phase,

w are weighting values determined during the training phase, and

b is the number of exponent sets in the sequence comprising said data.

28. A method as claimed in claim 27, wherein the selection stage comprises selecting

- 27 -

an output y_a of the first neural layer such that $y_a = 1$ and a is maximum for $1 < a < b$.

29. A method as claimed in claim 28, wherein the plurality of exponent coding strategies comprises strategies $S_1, S_2, \dots, S_c$, where $c \leq b$, corresponding to respective
5 differential coding limits 1, 2, $\dots$, c .

30. A method as claimed in claim 29, wherein the exponent coding strategy S_γ assigned to said first exponent set E_i is selected according to

$$\gamma = \max[\min(a+1, \sigma(E_i)), 1]$$

10 where $\sigma(E_i) = \text{floor}((\sum_j \|e_{i,j} - e_{i,j-1}\|/n) + 0.5)$.

31. A digital audio encoder in which audio data is transformed into coefficients having mantissas and exponents arranged in a sequence of sets, having:

a first variation processor coupled to receive the exponents of sets from said
15 sequence and determine a first variation of exponent values between a first set and a plurality of subsequent sets in the sequence;

a second variation processor coupled to receive the exponents of said first set and determine a second variation between consecutive exponent values within said first set;
and

20 a neural network processor coupled to receive said first and second variations and select and assign an exponent coding strategy to said first set from a plurality of coding strategies on the basis of said first and second variations.

32. An audio encoder as claimed in claim 31, wherein each of the plurality of coding
25 strategies correspond to a different differential coding limit.

33. An audio encoder as claimed in claim 31 or 32, wherein the neural network processor also selects and assigns an exponent coding strategy to at least one of the subsequent sets.

- 28 -

34 An audio encoder as claimed in claim 33, wherein the exponent coding strategy assigned to the at least one subsequent sets is an exponent re-use strategy.

35. An audio encoder as claimed in claim 31, wherein the neural network processor includes a weighted routing stage in which said first variation values are weighted according to predetermined weighting values and routed to inputs of a first neural layer, a selection stage in which an output of the first neural layer is selected, and an output processing stage in which a coding strategy is assigned to said first exponent set based on the output of said selection stage and said second variation.

36. An audio encoder as claimed in claim 35, wherein the first variation processor is arranged to determine said first variation according to

$$\text{Adiff}(E_i, E_j) = (\sum_m |e_{i,m} - e_{j,m}|)/n$$

where Adiff is said first variation,

E_i is said first exponent set and E_j is a subsequent exponent set with $i > j$,

$$E_i = (e_{i,0}, e_{i,1}, e_{i,2}, \dots, e_{i,n-1}),$$

$$E_j = (e_{j,0}, e_{j,1}, e_{j,2}, \dots, e_{j,n-1}),$$

n is an integer representing the number of exponents in a said set of exponents,

$$m = 0, 1, 2, \dots, n-1.$$

37. An audio coder as claimed in claim 31 or 36, wherein the weighted routing stage of the neural network processor is arranged to determine

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{b-1} \end{bmatrix} = \Gamma \left(\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & w_{1,1} & 0 & \dots & 0 \\ 0 & w_{2,1} & w_{2,2} & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & w_{b-1,1} & w_{b-1,2} & \dots & w_{b-1,b-1} \end{bmatrix} \begin{bmatrix} 0 \\ z_1 \\ z_2 \\ \dots \\ z_{b-1} \end{bmatrix} - \begin{bmatrix} 0 \\ T_1 \\ T_2 \\ \dots \\ T_{b-1} \end{bmatrix} \right)$$

- 29 -

the operator $\Gamma[\bullet]$ is defined as:

$$\Gamma \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \dots \\ \dots \\ \gamma_{b-1} \end{bmatrix} = \begin{bmatrix} f(\gamma_0) \\ f(\gamma_1) \\ f(\gamma_2) \\ \dots \\ \dots \\ f(\gamma_{b-1}) \end{bmatrix}$$

where $f(\gamma)$ is +1 if $\gamma \geq 0$ else it is 0,

Y represents outputs of the first neural layer,

T are threshold values determined during a training phase,

5 w are weighting values determined during the training phase, and

b is the number of sets in the sequence.

38. An audio encoder as claimed in claim 37, wherein the selection stage comprises selecting an output y_a of the first neural layer such that $y_a = 1$ and a is maximum for
10 $i < a < b$.

39. A method as claimed in claim 38, wherein the plurality of exponent coding strategies comprises strategies $S_1, S_2, \dots, S_c$, where $c \leq b$, corresponding to respective differential coding limits 1, 2, $\dots, c$.

15

40. A method as claimed in claim 39, wherein the exponent coding strategy S_γ assigned for encoding exponents in said first set E_i is selected according to

$$\gamma = \max[\min(a+1, \sigma(E_i)), 1]$$

where $\sigma(E_i) = \text{floor}((\sum_j \|e_{i,j+1} - e_{i,j}\|/n) + 0.5)$.

20