

# APPARATUS FOR CONTROLLING QUANTIZING IN A VIDEO SIGNAL COMPRESSOR

This invention relates to the quantizing apparatus in a video signal compression system.

5

Video signal compression apparatus typically include circuitry to quantize codewords of partially compressed signal, causing many of the quantized codewords to have similar values, which condition is conducive to run length encoding. Thereafter the quantized codewords are typically run and variable length encoded to provide the signal in compressed format. The term "partially compressed signal" refers to signal provided by, for example, a discrete cosine transform, or DCT, processor. Video data provided by DCT processors occurs in blocks of, for example, 8 x 8 coefficients representing a matrix of 8 x 8 contiguous pixels. Typically the respective coefficients in a block are quantized by different amounts depending upon the relative importance of the respective coefficients. In quantizing a block of such coefficients a quantization matrix is utilized. This quantization matrix includes the quantizing values to be applied to the respective coefficients in a block. Quantization of the video data may be performed by applying the same matrix or table of quantization values to all blocks of coefficients (codewords). This is referred to herein as fixed quantization, even though the quantizing values in the matrix may be different. Since different frames of video signal represent different images, it will be readily appreciated that the amount of compressed video data resulting from fixed quantization will be variable from frame to frame, giving rise to variable bit rate, VBR, compression.

Certain signal transmission systems cannot efficiently accommodate VBR compressed signal, hence constant bit rate, CBR, compression systems have been developed. In the CBR systems the quantization of codewords is made adaptive to control the number of compressed data. Typically the volume of compressed data is monitored and the quantization values are changed in inverse proportion to the volume of compressed data in order to render the number of compressed bits per frame substantially constant. Adaptive quantizing systems also utilize tables of

| REPLACED SHEETS                                                                      | SG |
|--------------------------------------------------------------------------------------|----|
| <input checked="" type="checkbox"/> Description : paged ....1... to paged ...14..... |    |
| <input checked="" type="checkbox"/> Claim(s) : paged ....15... to paged ...16.....   |    |
| <input checked="" type="checkbox"/> Drawing(s) : paged ....17... to paged ...18..... |    |
| <input checked="" type="checkbox"/> Abstract : paged ...19... to paged ...20.....    |    |
| DATE OF RECEIPT .....                                                                |    |

This Sheet Has Been Replaced



quantizing values for quantizing the blocks of coefficients. Adaptivity is typically realized by simply scaling the quantizing values as they are output from the table for application to the respective coefficients of a block. Nominally all quantizing values  
5 provided from the quantization table are scaled by the same factor, but the scale factor may vary from block to block.

CBR compression systems tend to cause the image quality from frame to frame to vary, but constant bit rate is advantageous if the compressed signal is to be multiplexed with  
10 other signals for transmission. Alternatively, VBR compressed signal tends to provide substantially uniform image quality from frame to frame, but is more difficult to transmit in a multiplexed signal situation.

15 The present invention is a quantizing system which provides limits to the amount of data, but substantially retains image quality uniformity from frame to frame. This system includes a quantizer for quantizing coded video data and further apparatus for monitoring the amount of output data. Dependent  
20 upon the amount of output data being lesser or greater than a predetermined value, the quantizer is conditioned to operate in different modes.

#### BRIEF DESCRIPTION OF THE DRAWINGS

25 FIGURES 1 and 2 are block diagrams of alternative video signal compression systems including quantization apparatus embodying the present invention.

FIGURE 3 is a flow chart of the operation of the FORWARD ANALYSER shown in FIGURE 2.

30 FIGURE 4, 5 and 6 are tables of representative quantization values for frames of video signal compressed according to intraframe processing, forward interframe processing and bidirectional interframe processing respectively.

35 FIGURE 7 is a pictorial representation of a quantizing system.

FIGURE 8 is a flow diagram of an adaptive quantization system wherein the threshold value is changed on a frame/field basis.



The present invention will be described in the environment of a video signal compressor of the type which does block or macroblock type of signal compression/handling, such as, for example, an MPEG-like compression system. MPEG is a system standard developed by the Moving Picture Experts Committee of the International Organization for Standardization. An example of an MPEG-like system may be found in United States Patent No. 5,122,875. The MPEG-like systems perform motion compensated predictive compression on respective successive groups of frames, GOP's. Within a compressed GOP the frames are designated I, P or B depending upon whether they are compressed by intraframe processing, forward interframe processing and bidirectional interframe processing respectively. Because the different frame types are compressed differently, the volume of compressed data for the different frame types may be different, even for the same image, hence different quantization values are appropriate for the different compressed frame types.

Compressed image data in an MPEG-like system is arranged in macroblocks which represent a matrix of e.g., 16 x 16 contiguous pixels. Respective macroblocks are configured of four 8 x 8 blocks of luminance data and two 8 x 8 blocks of chrominance data. Each 8 x 8 block of chrominance data represents the entire 16 x 16 matrix of pixels. The macroblocks are arranged in slices, with each slice representing, for example, 16 horizontal lines of video signal. One frame of 496 active lines can therefor be represented by 31 slices. MPEG-like systems typically quantize data on a macroblock basis.

Quantization in the present invention is nominally performed using fixed quantization values as long as the total coded data does not exceed a predetermined amount. In one embodiment, if it does then only the macroblocks of data that exceed certain limits are adaptively quantized, with all other macroblocks undergoing fixed quantization. In an MPEG-like compression environment, a maximum codeword target value is established for a GOP. This maximum target  $R_{GOP}$  is distributed amongst the respective frames in the GOP according to the compression process utilized for the respective frames. The  $R_{GOP}$



is reflected in global quantization factors  $Q_I$ ,  $Q_P$ ,  $Q_B$  used to modify the quantization values in the respective quantization matrices.

The total number of bits,  $R$ , used to code a GOP at a selected image quality is estimated from the relation;

5

$$R = r_I + \left(\frac{N}{M} - 1\right)r_P + \left(N - \frac{N}{M}\right)r_B \quad (1)$$

where  $r_I$ ,  $r_P$ ,  $r_B$  are the average nominal bit rates for VBR encoding intraframe, forward and bidirectional interframe predicted frames respectively,  $N$  is the number of frames in a GOP, and  $M$  is the number of B frames between two P frames. At system start up nominal values may be used for  $r_I$ ,  $r_P$ ,  $r_B$  in the first GOP, and thereafter the values from the prior GOP may be utilized until the first I, P and B frames are encoded for the current GOP. If the estimate  $R$  is less than  $R_{GOP}$ , the GOP is quantized with fixed quantization selected to produce a predetermined image quality. If it is not, a target bit allocation  $R_j$  is computed for each frame type as;

20

$$R_j = \frac{R_{GOP}}{W_j} \quad j = \{I, P, B\} \quad (2)$$

where

$$W_j = \begin{cases} 1 + \left(\frac{N}{M} - 1\right)\frac{1}{k_1} + \left(N - \frac{N}{M}\right)\frac{1}{k_2} & \text{for } j = I \\ k_1 + \left(\frac{N}{M} - 1\right) + \left(N - \frac{N}{M}\right)\frac{k_1}{k_2} & \text{for } j = P \\ k_2 + \left(\frac{N}{M} - 1\right)\frac{k_2}{k_1} + \left(N - \frac{N}{M}\right) & \text{for } j = B \end{cases} \quad (3)$$

25 and

$$k_1 = \frac{r_I}{r_P}, \quad k_2 = \frac{r_I}{r_B} \quad (4)$$

Each frame is first compressed using fixed quantization, and the total number of coded bits,  $r_j$ , for the frame is counted. After this first compression pass an excess rate fraction,  $xs\text{-frac}$ , is computed according to the equation;

30

$$xs\text{-frac} = \frac{r_j - R_j}{r_j} \quad j = I, P \text{ or } B \quad (5)$$



In addition a threshold value, ThV, is calculated from;

$$\text{ThV} = E[\text{mb}_{\text{size}}] + \text{StD}[\text{mb}_{\text{size}}]$$

5 where  $E[\text{mb}_{\text{size}}]$  is the average number of compressed bits per macroblock in the frame and  $\text{StD}[\text{mb}_{\text{size}}]$  is the standard deviation in the number of bits per macroblock. The value ThV may be computed using the volume of data in respective macroblocks either prior to or after quantization. That is ThV may be a  
10 function of unquantized data or quantized data.

If the value of xs-frac is less than zero, the number of coded bits is within the set limits, and the frame is coded in normal VBR mode with the fixed quantization used in the first compression pass. If it is not, a second compression pass is  
15 invoked. In the second pass, fixed quantization is used for all macroblocks whose size is less than ThV. For the remaining macroblocks a larger quantization factor is selected. This larger quantization factor may be selected according to a formula and/or dependent upon, for example, the difference between  $\text{mb}_{\text{size}}(i)$   
20 and ThV. In this instance the possible difference values may be divided into ranges, and a quantization factor increment assigned to each range. Assume that the potential difference values are divided into three ranges which encompass 50, 30 and 20 percent of the possible values, with the 50, 30 and 20 percent ranges  
25 including the lowest to the highest values respectively. The 50, 30 and 20 percent ranges may be assigned increment values of 1, 2 and 3 respectively. Thus if a  $\text{mb}_{\text{size}}(i)$  exceeds ThV by an amount represented by the second range, the current quantization factor for that block is incremented by 2.

30 Alternatively the use of empirical lookup tables of quantization factors for macroblocks exceeding the threshold has been found to produce good results. FIGURES 6, 7, and 8 represent such empirical lookup tables. In selecting a larger quantization factor two quantities are used to enter the empirical  
35 table. These two quantities are the current quantization factor and the variable xs-frac. The output from the table is an updated quantization factor which is used to quantize the current macroblock. Different tables are used for the different encoded



frame types. The tables in FIGURES 6, 7 and 8 correspond to values developed for use with I, P and B frames respectively. It should be noted that certain macroblocks in P and B encoded frames may be intraframe encoded. In this instance the I table  
 5 will be used for such I-type macroblocks found in P or B frames. In these tables the values labelled OLD  $q_i$  are the current quantization factors (used to address the table), the values labelled NEW  $q_i$  are the new quantization factors output from the table, and the values shown internal to the table are the values  
 10 xs-frac multiplied by a predetermined amount, e.g. one hundred, which xs-frac values are also used to address the respective tables.

It has been found to be beneficial to update the threshold value ThV on a macroblock by macroblock basis. To  
 15 this end a running variable  $\alpha$  is computed as respective macroblocks are processed. Successive threshold values  $ThV_n$  are calculated according to:

$$ThV_n = \alpha_n(ThV) \quad (6)$$

The variable  $\alpha_n$  is calculated from the equation:

$$20 \quad \alpha_{n+1} = \alpha_n - \gamma_n K_n \quad (7)$$

Successive values of  $\alpha$  are reduced by the quantity  $\gamma_n K_n$ . The factor  $\gamma_n$  is utilized to insure that the quantity  $\gamma_n K_n$  does not grow too rapidly, that is to insure that  $\alpha$  is not reduced too rapidly.

Both factors are a function of the accumulated number of bits  
 25 occurring in the preceding macroblocks. The factor  $K_n$  is given by:

$$K_n = s \left( \frac{xs\text{-}frac - Cut\text{-}frac_n}{xs\text{-}frac} \right) \quad (8)$$

where  $s$  is a constant such as 0.1 and the variable Cut-frac is  
 30 computed from;

$$Cut\text{-}frac_n = \frac{\sum_{i=1}^n bits\text{-}cut(i)}{\sum_{i=1}^n mb_{size}(i)} \quad (9)$$



The term bits-cut refers to the number of coded bits by which the preceding macroblocks have been reduced as a result of adaptive quantization of the macroblocks exceeding ThV.

The variable  $\gamma_n$  is computed using the equation;

5

$$\gamma_n = \frac{\sum_{i=1}^n mb_{size}(i)}{N \cdot \sum_{i=1}^n mb_{size}(i)} \quad (10)$$

and bears a very small value until a significant number of the busier macroblocks in a frame have been processed.

Alternatively  $\gamma_n$  may be assigned fixed values for different

10 groupings of macroblocks. This value may be zero, 0.1 and 0.25 for the first occurring one third of the macroblocks, the second occurring one third of the macroblocks and the third occurring one third of the macroblocks respectively. Note that the values calculated in equations (6) through (10), after processing the  $n^{th}$  macroblock, are applied in processing the  $n + 1^{th}$  macroblock.

15 The foregoing process may be iterated several times before the total coded bits satisfy equation (5). For successive iterations the variable  $xs\_frac$  is weighted in a direction to provide larger quantization factors from the empirical tables. A working  
20 table of quantization factors is kept, in which the quantization factor for each macroblock is stored. As quantization factors for respective macroblocks are changed the corresponding quantization factor in the table is updated. After the necessary iterations are completed the respective frame is recoded using the  
25 quantization factors stored in the table for quantizing the respective macroblocks.

FIGURE 1 illustrates a general form of a system for practicing the invention. This particular system assumes MPEG encoding, but it will be recognized that any encoder which  
30 operates on a block basis may incorporate the invention. Video signal to be encoded is applied to a bus 2, which is coupled to the input of an MPEG encoder 4 and a memory device 7. The encoder 4 is coupled to a forward analyzer 6, and communicates thereto information related to the encoding process. This information



primarily comprises the current frame type, the current macroblock type, the current quantization factor for respective macroblocks and the number of bits encoded. From this data the forward analyzer performs the computations of equations (1) - (10) and generates new quantization factors which are loaded in the aforementioned working table. The forward analyzer 6 forces sufficient coding iterations of each frame to adapt the quantization factors such that equation (5) is satisfied.

The video data stored in the memory 7 is coupled to a second MPEG encoder 8, a frame at a time, after quantization factors for the corresponding frame have been determined by the forward analyzer. Concurrently with the stored frame being coupled to the second MPEG encoder the forward analyzer couples the quantization factors from the working table of quantization factors selected for that corresponding frame. The MPEG encoder 8 uses these quantization factors for encoding the corresponding frame of video data.

A system like that illustrated in FIGURE 1 can process more data per unit time but tends to be hardware intensive. A less hardware intensive system is illustrated in FIGURE 2, wherein certain parts of the apparatus are operated in time division multiplexed fashion to perform the respective operations of forward analysis and encoding.

In FIGURE 2 the elements 10 to 22 comprise a typical motion compensated predictive encoder. There are three parallel processing paths shown, respective ones of which process the luminance and two chrominance video components. The motion compensation apparatus 20 operates on the luminance signal to generate motion vectors, which vectors are used in the three processing paths. A common controller 21 communicates with the respective processing circuitry and directs the general operation of the encoder. The operation of motion compensated predictive encoders is well known and will not be described in detail herein.

The apparatus of FIGURE 2 also includes a forward analyzer 25 including a storage element 26, a processor 27, and a counter 28. The processor 27 controls the storage element 26, the counter 28, and communicates with the controller 21 to interrupt processing to allow iteration of quantization adaption. In FIGURE



2, a frame of video signal is applied to the bus 10 and compressed (encoded) using fixed quantization. As the respective blocks of partially compressed data are generated by the DCT processor, 13, these blocks are stored in the memory 26 to be used for  
5 subsequent iterations of quantization factor generation. The quantized data from the quantizing element 14, is applied to a run-length and variable length encoder 15. Run and variable length encoded data from the encoder 15 and motion vectors from element 20 are coupled to a formatter 22. The formatter 22  
10 generates header data and arranges the blocks of data from the encoder 15, the motion vectors, and the header data into macroblocks, slices, frames and GOP's etc.

Coded data from either the encoder 15 or the formatter 22 is counted by the counter 28 to determine the data  
15  $r_j$  for the encoded frame. Whether the data from the formatter 22 or the encoder 15 is counted depends upon whether the amount of data corresponding to motion vectors and/or header data is relatively constant and predictable for the respective frame types. If it is, then data may be counted at the output of either device,  
20 and if it is not then the data to be counted must be derived from the output of the formatter 22.

The processor 27 is programmed to communicate with the system controller 21 to acquire the bit rate  $R_{GOP}$ , and at the end of each encoding pass to acquire the count of encoded bits  
25 from the counter 28. Using this data the processor 27 performs the quantization factor adaption. This process will be described with the aid of the flowchart of FIGURE 3. The system is initialized [50] by applying user parameters to the system controller 21, which parameters indicate for example the type of video source  
30 material, the desired coding resolution, the desired image quality etc. The system controller is substantially a state machine and these parameters establish the starting and selected states. If such user parameters are not entered, the system controller loads a set of default parameters, one of which is  $R_{GOP}$ . The  $R_{GOP}$  value  
35 is communicated to the processor 27, and nominal values for  $r_I$ ,  $r_P$ ,  $r_B$  are accessed from, for example, read only memory. Using these values the processor 27 calculates the respective  $R_j$  values.

Next the system controller accesses [51] the first frame



of data. The respective frames are designated I, P, B by the controller 21 according to the relative position in the sequence of input frames that particular frames occur. The frame is encoded [52] using a quantization table and a global quantization factor  $Q_j$  selected to provide a predetermined image quality. During this process the blocks of data provided from the DCT apparatus are stored in the memory 26. As respective macroblocks are encoded, a count value from the counter 28 associated with the macroblock is also stored. This count value may be the cumulative count from the beginning of the frame, or a value indicating the amount of coded data for the respective macroblock. The latter value may be obtained by storing the output from the counter for one macroblock period and subtracting the respective stored count value from the cumulative count value associated with the next successive macroblock.

After the frame is compressed, the total data,  $r_j$ , for the encoded frame is acquired from the counter 28 and the variable  $xs\text{-}frac$  is calculated [53] according to equation (5). In addition the threshold value  $ThV$  is also calculated. The variable  $xs\text{-}frac$  is tested [54] to determine if it is positive. If it is positive, the encoded data for the frame are less than the  $R_j$ , and no quantization adjustments are required. The global quantization factor is loaded [55] into all of the locations of a working quantization table ( $q\text{ MEM}$ ) within the processor 27.

At this point there are at least two system alternatives. The system can be instructed to re-encode the frame using the global quantization factor, and to output the encoded data to further circuitry such as a rate buffer (not shown). In this instance such further circuitry would be inhibited from accepting data from the formatter until after the decision made in step [54]. In an alternative system, the formatter may be coupled to a memory which stores the lastmost frame of encoded data. Then if the condition of step [54] is true the encoded data in such memory is output, obviating re-encoding the frame.

If  $xs\text{-}frac$  is not less than zero [54], the quantization adaption process is initiated [56]. Variables  $\alpha$  and  $\beta$  are set [57] to 1, and the variable  $xs\text{-}frac$  is scaled [60] by  $\beta$ . In the first iterative pass, since  $\beta$  is equal to 1,  $xs\text{-}frac$  is unchanged. However



in successive iterations  $\beta$  is incremented hence xs-frac is increased providing two results. The first is that quantization values selected from the respective tables will be greater. The second is that the variable  $\alpha$  will change more rapidly.

5           After xs-frac is scaled, the first macroblock in the frame is accessed [66] from the memory 26, and the macroblock is tested [67] to determine if the amount of coded data for that macroblock exceeds the current threshold value. The current threshold value is the original threshold value ThV scaled by the  
10       variable  $\alpha_n$ . If the amount of coded data for the current macroblock exceeds the current threshold value, the appropriate empirical table is addressed [68] using the current quantization factor and the value xs-frac to secure a new quantization factor for that specific macroblock. That quantization factor is loaded  
15       [77] in the working quantization table (q MEM in the processor 27) , at an address location associated with the respective macroblock. The variable bits-cut is updated [69] and the variable Cut-frac is updated [70] from data stored in the memory 26 and data provided from the counter 28.. The variables  $\gamma_n G_n$   
20       and  $\alpha_n$  are updated according to equations (7, 8 and 10) at step [71]. A check is made [72] to determine if the last macroblock has been processed. If not, the macroblock address is incremented [73] and the next macroblock of data is accessed from the memory 26.

25           If the last macroblock has been processed [72] the variable xs-frac is tested [74] against the variable Cut-frac to determine if sufficient data reduction has been attained. If it has, then the system is instructed to re-encode the frame with the quantization factors in the working quantization table, and then to  
30       get the next frame [51].

          If one or more iterations are required to establish the requisite quantization factors, it is necessary to re-encode the corresponding frame of data because varying the quantization factors will affect the predicted frames provided by the predictor  
35       19. During re-encoding of the frame, however it is not necessary to recalculate the motion vectors, since they should not be affected. This presumes that the formatter 22 includes sufficient



memory to store the motion vectors so that they need not be recomputed.

Returning to step [67], if the test is not true, no action need be taken regarding the tested macroblocks, and the process jumps to step [70] to update Cut-frac in preparation for examining the next macroblock. At step [74] if the test is true a further iteration is initiated.  $\beta$  is incremented [75] (by a small amount, for example 0.1), and the macroblock address is reset [76] to the first macroblock of a frame and  $\alpha$  is reset to 1.

During each quantization iteration it is not necessary to re-encode the frame of data. The only data necessary for these calculations is the macroblock data from the first encoding pass, which data is stored in the memory 26. It is possible however to dispense with an extra memory 26 and re-encode all of the data each iteration if time and power constraints permit.

The actual quantization process is elucidated with respect to FIGURE 7. In FIGURE 7 the block 83 represents a block of coefficient data from a given macroblock. The respective coefficients  $C_i$  (corresponding to discrete cosine transform coefficients for example) are arranged in a predetermined order such as according to the ascending index  $i$ . The block designated 80 represents a matrix of quantization values  $q_i$ . These quantization values are selected so as to provide greater and lesser quantization to coefficients in the block 83 which are of lesser and greater importance, respectively, to image reproduction. Thus when coefficients  $c_1, c_2, \dots, c_8$  etc. are applied to the quantizer 81, quantization values  $q_1, q_2, \dots, q_8$  etc. are concurrently accessed and applied to perform the quantization. However the values  $q_i$  are indirectly applied to the quantizer 81 via a variable scaling circuit 82. The respective quantization values are scaled by weighting factors stored in the working quantization memory 84.

The respective blocks in the memory 84 correspond to the quantization factor to be applied to a particular macroblock in a frame. For fixed quantization the memory 84 is loaded with one value which is equal to a global quantization factor related to the particular frame type i.e.  $Q_I, Q_P$  or  $Q_B$ . The  $Q$ 's in the blocks of the memory 84 represent the global quantization factor. The  $X$ 's



represent updated quantization factors resulting from the foregoing quantization adaption process. These updated values are inserted into memory 84 at step [77] in FIGURE 3.

As blocks of coefficients 83 are applied to the quantizer 81, the corresponding quantization factor from the memory 84 is applied to the scaling circuit 82 to produce quantization values  $Q_j(q_i)$  or  $X(q_i)$ , which are coupled to the quantizer 81. The quantizer 81 may be of the form which quantizes the respective coefficients according to the functions:

10

$$\begin{aligned} Q_{Ci} &= (16 * c_i + 8) // (2 * X_{qi}); & c_i > 0 \\ Q_{Ci} &= (16 * c_i - 8) // (2 * X_{qi}); & c_i < 0 \\ Q_{Ci} &= 0; & c_i = 0 \end{aligned}$$

15 where  $Q_{Ci}$  is the quantized coefficient,  $X_{qi}$  represent the quantization values provided from the scaling circuit 82, the asterisk \* denotes multiplication and // specifies inter division.

In the discussion above the variable  $xs\text{-}frac$  was changed or updated on a frame basis. This variable may be updated on a shorter basis such as every slice or every W macroblocks.

FIGURE 8 illustrates a flowchart for a simplified adaptive quantizing system. In this system the threshold value  $ThV_n$  is changed on a frame basis rather than a macroblock basis. Changing the threshold on a frame basis requires a more accurate estimate of the initial threshold value, but provides more uniform image quality. Note the system of FIGURE 8 does not require the calculation of the variables defined in equations (8) and (10). In this instance  $\alpha$  is decremented by a constant each iteration. In FIGURE 8 the value of BITS-CUT is given by

30

$$BITS-CUT = \sum_{i=1}^n bits-cut(i)$$

where  $bits-cut(i)$  is equal to the number of bits a particular macroblock is reduced from the initial encoding pass.

A major feature of this invention is that adaptive quantization is performed on only certain blocks (macroblocks) in each frame. These blocks tend to be blocks which contain a greater amount of image detail than average blocks within a



respective frame. Adaptively quantizing these "busier" blocks has the least deleterious affect on image quality, hence subjective image quality is not substantially affected by the process.

At present the invention has not been optimized since  
5 sufficient statistics for compressed video data were not available to the inventors. However, it is very probable, when such statistics are available, that a threshold value function which is related to the amount of data in respective macroblocks will be determined, which will allow quantization factor determination  
10 from the data generated during the first compression pass with fixed quantization.

This Sheet Has Been Replaced



5

CLAIMS:



1. Apparatus for quantizing video data occurring as blocks of codewords, comprising:  
a source (13) of said blocks of codewords;  
characterized by:  
means (27) for providing a threshold value;  
comparing means (27,28) for determining whether the volume of data in respective blocks is greater or lesser than said threshold value; and  
means for quantizing (14) blocks of codewords whose volume of data is less than said threshold according to predetermined quantizing values, and for quantizing blocks of codewords whose volume of data exceeds said threshold value with quantizing values arranged to provide greater quantization than said predetermined quantizing values.
2. The apparatus set forth in claim 1 characterized in that said means for providing a threshold value includes;  
means for statistically analyzing a predetermined plurality of said blocks of codewords; and  
means responsive to statistics of said predetermined plurality of blocks for generating said threshold value.
3. The apparatus set forth in claim 2 characterized in that said means for statistically analyzing provides a value representing the average volume of data per block, and the standard deviation of the volume of data from said average and said means responsive to statistics generates said threshold value as a function of said average and said standard deviation.
4. The apparatus set forth in claim 2 characterized in that said blocks of codewords are arranged in macroblocks and said apparatus further includes means for altering said threshold value on a macroblock by macroblock basis.

This Sheet Has Been Replaced





5. The apparatus set forth in claim 2 characterized in that said blocks of codewords are arranged in macroblocks and said apparatus further includes means for periodically altering said threshold value at predetermined numbers of macroblocks.

6. The apparatus set forth in claim 1 characterized in that said means for quantizing comprises:

a source of a matrix of quantizing values preselected for quantization of particular codewords in respective blocks;  
means for weighting said matrix of quantizing values;  
and

means for applying a predetermined weighting factor to said means for weighting for respective blocks of codewords whose volume of data is less than said threshold value, and for applying adaptively generated weighting factors for respective blocks of codewords whose volume exceeds said threshold value.

7. The apparatus set forth in claim 6 characterized in that said means for quantizing further includes;

first memory means having preselected weighting factors stored at respective address locations;

second memory means having address locations corresponding to respective blocks of codewords, said address locations containing weighting factors previously applied for quantizing respective blocks; and

means for addressing said second memory responsive to weighting factors stored in said second memory means

This Sheet Has Been Replaced

