AU9174763

(12) PATENT ABRIDGMENT (11) Document No. AU-B-74763/91
(19) AUSTRALIAN PATENT OFFICE (10) Acceptance No. 653715

(54) Title
ADAPTIVE BLOCK SIZE IMAGE COMPRESSION METHOD AND SYSTEM

International Patent Classification(s)
(51)[5] **H04N 007/133** **G06K 009/36** **H04N 001/415**

(21) Application No. : **74763/91** (22) Application Date : **26.02.91**

(87) PCT Publication Number : **WO91/13514**

(30) Priority Data

(31) Number	(32) Date	(33) Country
487012	**27.02.90**	**US UNITED STATES OF AMERICA**

(43) Publication Date : **18.09.91**

(44) Publication Date of Accepted Application : **13.10.94**

(71) Applicant(s)
QUALCOMM, INC.

(72) Inventor(s)
CHONG U LEE

(74) Attorney or Agent
R K MADDERN & ASSOCIATES , 345 King William Street, ADELAIDE SA 5000

(56) Prior Art Documents
US 4924309
US 4504860
US 4293920

(57) Claim

1. In an image compression system, a sub-system for generating from a block of input pixel data a corresponding composite block of discrete cosine transform (DCT) data, comprising:

transform means for, receiving an input block of pixel data, performing a discrete cosine transform (DCT) operation on said block of pixel data and on at least one predetermined level of constituent sub-blocks thereof, and providing corresponding block and sub-blocks of DCT coefficient values;

block size assignment means for, receiving said block and each sub-block of DCT coefficient values, determining for each block and each corresponding group of sub-blocks of DCT coefficient values a bit count value corresponding to a number of bits required to respectively encode said block and each corresponding group of sub-blocks of DCT coefficient values according to a predetermined coding format, determining from said bit count values ones of said block and corresponding group of sub-blocks of DCT coefficient values requiring a lesser number of bits to encode according to said coding format, and providing a corresponding selection value; and

.../2

selection means for, receiving said selection value and said block and each sub-block of DCT coefficient values, selecting ones of block and sub-blocks of DCT coefficients corresponding to said selection value, and providing an output of a corresponding composite block of DCT coefficient values formed from said selected ones of block and sub-blocks of DCT coefficient values.

OPI DATE 18/09/91 APPLN. ID 74763 / 91
AOJP DATE 24/10/91 PCT NUMBER PCT/US91/01350
INTERN N TREATY (PCT)

(51) International Patent Classification 5 : H04N 1/415, G06K 9/36

A1

(11) International Publication Number: WO 91/13514

(43) International Publication Date: 5 September 1991 (05.09.91)

(21) International Application Number: PCT/US91/01350

(22) International Filing Date: 26 February 1991 (26.02.91)

(30) Priority data:
487,012 27 February 1990 (27.02.90) US

(71) Applicant: QUALCOMM, INC. [US/US]; 10555 Sorrento Valley Road, San Diego, CA 92121-1617 (US).

(72) Inventor: LEE, Chong, U. ; 5238 Camino Playa Malaga, San Diego, CA 92124 (US).

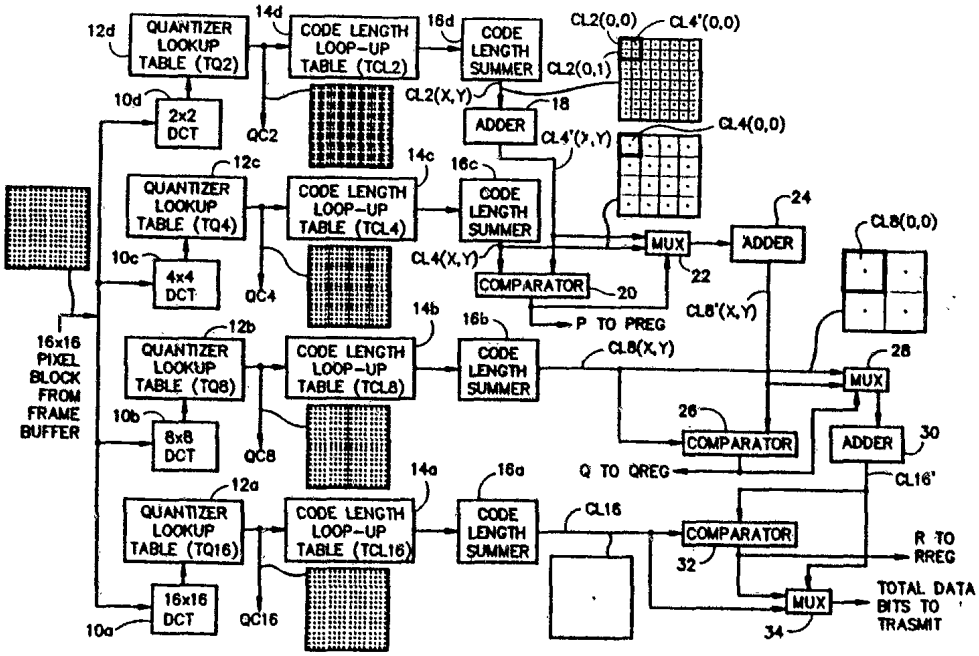
(74) Agent: MILLER, Russell, B.; Qualcomm, Inc., 10555 Sorrento Valley Road, San Diego, CA 92121-1617 (US).

(81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FI, FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), NO, SE (European patent).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

653715

(54) Title: ADAPTIVE BLOCK SIZE IMAGE COMPRESSION METHOD AND SYSTEM



(57) Abstract

An image compression system and method for compressing image data for transmission. Each block and corresponding sub-blocks of pixel data is subjected to a discrete cosine transform (DCT) operation (10, 12). Varying levels of sub-blocks (QC16, QC8, QC4, QC2) of resulting corresponding transform coefficients are selected for construction into a composite transform coefficient block (QC) corresponding to each input block of pixel data. The selection of transform coefficient block size for the composite block is determined by a comparison process between transform block and sub-block coding efficiency (20, 26, 32). The composite block is variable length coded (56) to further reduce bit count in the compressed data.

ADAPTIVE BLOCK SIZE IMAGE COMPRESSION METHOD AND SYSTEM

BACKGROUND OF THE INVENTION

I. Field of the Invention

The present invention relates to image processing. More to particularly, the present invention relates to a novel and improved image signal compression scheme utilizing adaptively sized blocks and sub-blocks of encoded discrete cosine transform (DCT) coefficient data.

II. Description of the Related Art

In the field transmission and reception of television signals, various improvements are being made to the NTSC (National Television Systems Committee) System. Developments in the field of television are commonly directed towards a high definition television (HDTV) System. In the development of HDTV, system developers have merely applied the Nyquist sampling theorem and low pass filtering design with varying degrees of success. Modulation in these systems amounts to nothing more than a simple mapping of an analog quantity to a value of signal amplitude or frequency.

It has most recently been recognized that it is possible to achieve further improvements in HDTV systems by using digital techniques. Many of the proposed HDTV transmission formats share common factors. These systems all involve digital processing of the video signal, which necessitates analog-to-digital (A/D) conversion of the video signal. An analog transmission format is then used thereby necessitating conversion of the digitally processed picture back to analog form for transmission.

The receiver/processor must then reverse the process in order to provide image display. The received analog signal is therefor digitized, stored, processed and reconstructed into a signal according to the interface format used between the receiver/processor and the HDTV display. Furthermore the signal is most likely converted back to analog form once more for display. It is noted however that the proposed HDTV formats

utilize digital transmission for transmission of control, audio and authorization signals.

Many of the conversion operations mentioned above, however, may be avoided using a digital transmission format which transmits the processed picture, along with control, audio and authorization signals ,using digital modulation techniques. The receiver may then be configured as a digital modem with digital outputs to the video processor function. Of course, the modem requires an A/D function as part of operation, but this implementation may only require a 4-bit resolution device rather than the 8-bit resolution device required by analog format receivers.

Digital transmission is superior to analog transmission in many ways. Digital transmissions provide efficient use of power which is particularly important to satellite transmission and in military applications. Digital transmissions also provides a robustness of the communications link to impairments such as multipath and jamming. Furthermore digital transmission facilitates ease in signal encryption, necessary for military and many broadcast applications.

Digital transmission formats have been avoided in previous HDTV system proposals primarily because of the incorrect belief that they inherently require excessive bandwidth. Therefore in order to realize the benefits of digital transmission, it is necessary to substantially compress the HDTV signal. HDTV signal compression must therefor be achieved to a level that enables transmission at bandwidths comparable to that required by analog transmission formats. Such levels of signal compression coupled with digital transmission of the signal will enable a HDTV system to operate on less power with greater immunity to channel impairments.

It is therefore an object of the present invention to provide a novel and improved method and system for compressing HDTV signals that will enable digital transmission at bandwidths comparable to that of analog transmissions of convention TV signals.

SUMMARY OF THE INVENTION

The present invention is a novel and improved system and method for compressing image data for transmission and for reconstruction of the image data upon reception. The image compression system includes a sub-system for generating from a block of input pixel data a corresponding composite block of discrete cosine transform data optimized for encoding for a minimized transmission data rate.

The sub-system of the present invention comprises transform means for receiving an input block of pixel data and for performing a discrete cosine transform (DCT) operation on the block of pixel data and on at least one predetermined level of constituent sub-block thereof. The transform means provides an output of corresponding block and sub-blocks of DCT coefficient values.

Block size assignment means is included in the sub-system for receiving the block and each sub-block of DCT coefficient values and for determining for the block and each corresponding group of constituent sub-blocks of DCT coefficient values a bit count value corresponding to a number of bits required to respectively encode the block and each corresponding group of constituent sub-blocks of DCT coefficient values according to a predetermined coding format. The block assignment means is further for determining from the bit count values ones of the block and group of constituent sub-blocks of DCT coefficient values requiring a lesser number of bits to encode according to the coding format, and providing an output of a corresponding selection value.

The sub-system further includes selection means for receiving the selection value and the block and each sub-block of DCT coefficient values and for selecting ones of block and sub-blocks of DCT coefficient values corresponding to the selection value. The selection means provides an output of a corresponding composite block of DCT coefficient values formed from the selected ones of block and sub-locks of DCT coefficient values.

The system embodying the sub-system further comprises ordering means for receiving the composite block of DCT coefficient values and ordering the composite block of coefficient values according to a

predetermined ordering format. The ordering means provides an output of the ordered DCT coefficient values.

The ordered DCT coefficient values output from the ordering means are received by an encoder means that codes the ordered DCT coefficient values according to the predetermined coding format. The encoder means provides an output of the coded ordered DCT coefficient values.

The coded values are received by an assembler means which also receives the selection value provided by the selection means. The assembler means combines the selection value and the coded values as a coded image value representative of the input block of pixel data. The coded image value is of a reduced bit count with respect to a bit count of the input block of pixel data. The assembler means provides an output of the coded image value for transmission.

The present invention also provides for a novel and improved method for reconstructing from each received coded image value corresponding a block pixel of pixel data. The present invention further envisions a novel and improved method for compressing an image signal as represented by a block of pixel data and for reconstructing the image signal from the compressed image signal.

BRIEF DESCRIPTION OF THE DRAWINGS

The features, objects, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

Figure 1 is a block diagram illustrating the processing elements of the adaptive block size image compression system for providing DCT coefficient data and block size determination;

Figure 2 is a block diagram illustrating the processing elements for selecting block sizes of DCT coefficient data so at to generate a composite block of DCT coefficient data and the encoding of the composite block for transmission;

Figures 3a and 3b respectively illustrate exemplarily register block size assignment data and the block selection tree corresponding thereto;

Figures 4a and 4b are graphs respectively illustrating in graphical form the selected block zig-zag scan serialization ordering sequence within the sub-blocks and between sub-blocks for an exemplary composite block of DCT coefficient data whose block size selection was made according to the block size assignment data of Figure 3a;

Figures 5a-5d respectively illustrate in graphical form an alternate zig-zag scan serialization format;

Figure 6 is a block diagram illustrating the processing elements for decoding and reconstructing an image from a received signal generated by the processing elements of Figures 1 and 2;

Figure 7 is a flow chart illustrating the processing steps involved in compressing and coding image data as performed by the processing elements of Figures 1 and 2; and

Figure 8 is a flow chart illustrating the processing steps involved in decoding and decompressing the compressed signal so as to generate pixel data.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In order to facilitate digital transmission of HDTV signals and enjoy the benefits thereof, it is necessary to employ some form of signal compression. In order to achieve such high definition in the resulting image, it is also important that high quality of the image also be maintained. The discrete cosine transform (DCT) techniques have been shown to achieve a very high compression. One such article which illustrates the compression factor is that entitled "Scene Adaptive Coder" by Wen-Hsiung Chen et al., IEEE Transactions on Communications, Vol. Com-32, No. 3, March, 1984. However, the quality of reconstructed pictures is marginal even for video conferencing applications.

With respect to the DCT coding techniques, the image is composed of pixel data which is divided into an array of non-overlapping blocks, NxN in size. Strictly for black and white television images each pixel is represented by an 8-bit word whereas for color television each pixel may be represented by a word comprised of up to 24-bits. The blocks in which the image is divided up to is typically a 16 x 16 pixel block, i.e. N =16.

A two-dimensional NxN DCT is performed in each block. Since DCT is a separable unitary transformation, a two-dimensional DCT is performed typically by two successive one-dimensional DCT operations which can result in computational savings. The one-dimensional DCT is defined by the following equation:

$$X(k) = \frac{2C(k)}{N} \sum_{n=o}^{N-1} x(n) \cos \frac{p(2n+1)k}{2N} \qquad (1)$$

where $C(0) = \frac{1}{\sqrt{2}}$;

and

$C(k) = 1$ for $k = 1,2,3,\ldots N-1$.

For television images, the pixel values are real so that the computation does not involve complex arithmetic. Furthermore, pixel values are non-negative so that the DCT component, X(0), is always positive and usually has the most energy. In fact, for typical images, most of the transform energy is concentrated around DC. This energy compaction property makes the DCT such an attractive coding method.

It has been shown in the literature that the DCT approaches the performance of the optimum Karhunen-Loeve Transform (KLT), as evidenced by the article entitled "Discrete Cosine Transform" by N. Ahmed et al., IEEE Transactions on Computers, January 1974, pages 90-93. Basically, the DCT coding performs a spatial redundancy reduction on each block by discarding frequency components that have little energy, and by assigning variable numbers of bits to the remaining DCT coefficients depending upon the energy content. A number of techniques exist that quantize and allocate bits to minimize some error criterion such as MSE over the block. Typically the quantized DCT coefficients are mapped into a one-dimensional string by ordering from low frequency to high frequency. The mapping is done according to diagonal zig-zag mapping over the block of DCT coefficients. The locations of the zero (or discarded) coefficients are then coded by a run-length coding technique.

In order to optimally quantize the DCT coefficient, one needs to know the statistics of the transform coefficients. Optimum or sub-optimal

quantizers can be designed based on the theoretical or measured statistics that minimize the over-all quantization error. While there is not complete agreement on what the correct statistics are, various quantization schemes may be utilized, such as that disclosed in "Distribution of the Two-Dimensional DCT Coefficients for Images" by Randall C. Reininger et al., IEEE Transactions on Communications, Vol. 31, No. 6, June 1983, Pages 835-839. However, even a simple linear quantizer has been utilized which has provided good results.

Aside from deciding on a quantization scheme, there are two other methods to consider in order to produce the desired bit rate. One method is to threshold the DCT coefficient so that the small values are discarded and set to zero. The other technique is to linearly scale (or normalize) to coefficients to reduce the dynamic range of the coefficients after floating point to integer conversion for coding. Scaling is believed to be superior to thresholding in retaining both the subjective as well as objective signal to noise ratio quality. Therefore the main variable in the quantization process will be the coefficient scale factor which can be varied to obtain the desired bit rate.

The quantized coefficients usually are coded by Huffman codes designed from the theoretical statistics or from the measured histogram distribution. Most of the coefficients are concentrated around the low values so that Huffman coding gives good results. It is believed that Huffman codes generated from a measured histogram performs very close to theoretical limits set by the entropy measure. The location of the zero coefficients are coded by run-length codes. Because the coefficients are ordered from low to high frequencies, the runs tend to be long such that there is a small number of runs. However, if the number of runs in terms of length were counted, the short runs dominate so that Huffman coding the run-lengths reduces the bit rate even more.

An important issue that concerns all low bit-rate compression schemes is the effect of channel bit error on the reconstruction quality. For DCT coding, the lower frequency coefficients are more vulnerable especially the DC term. The effect of the bit error rate (BER) on the reconstruction quality at various compression rates has been presented in the literature. Such issues are discussed in the article entitled "Intraframe Cosine Transfer Image Coding" by John A. Roese et al., IEEE Transactions

on Communications, Vol. Com-25, No. 11, November 1977, Pages 1329-1339. The effect of BER becomes noticeable around 10^{-3} and it becomes significant at 10^{-2}. A BER of 10^{-5} for the transmission sub-system would be very conservative. If necessary, a scheme can be devised to provide additional protection for lower frequency coefficients, such as illustrated in the article " Hamming Coding of DCT-Compressed Images over Noisy Channels" by David R. Comstock et al., IEEE Transactions on Communications, Vol. Com-32, No. 7, July 1984, Pages 856-861.

It has been observed that most natural images are made up of blank relatively or slow varying areas, and busy areas such as object boundaries and high-contrast texture. Scene adaptive coding schemes take advantage of this factor by assigning more bits to the busy area and less bits to the blank area. For DCT coding this adaptation can be made by measuring the busyness in each transform block and then adjusting the quantization and bit allocation from block to block. The article entitled "Adaptive Coding of Monochrome and Color images" by Wen-Hsiung Chen et al., IEEE Transactions on Communications, Vol. Com-25, No.11, November 1977, Pages 1285-1292, discloses a method where block energy is measured with each block classified into one of four classes. The bit allocation matrix is computed iteratively for each class by examining the variance of the transform samples. Each coefficient is scaled so the desired number of bits result after quantization. The overhead information that must be sent are the classification code, the normalization for each block, and four bit allocation matrices. Utilization of this method has produced acceptable results at 1 and 0.5 bits per pixel.

Further bit rate reduction was achieved by Chen et al in the previously mentioned article "Scene Adaptive Coder" where a channel buffer is utilized to adaptively scale and quantize the coefficients. When the buffer becomes more than half full, a feedback parameter normalizes and quantizes the coefficients coarsely to reduce the bits entering the buffer. The converse happens when the buffer becomes less than half full. Instead of transmitting the bit allocation matrices, they run-length code the coefficient locations and Huffman code the coefficients as well as the run-lengths. Such an implementation has shown good color image reconstructions at 0.4 bits per pixel. Although these results look very good when printed, the simulation of the system shows many deficiencies.

When images are viewed under normal to moderate magnification smoothing and blocking effects are visible.

In the present invention, intraframe coding (two-dimensional processing) is utilized over interframe coding (three-dimensional processing). One reason for the adoption of intraframe coding is the complexity of the receiver required to process interframe coding signals. Interframe coding inherently require multiple frame buffers in addition to more complex processing circuits. While in commercialized systems there may only be a small number of transmitters which contain very complicated hardware, the receivers must be kept as simple as possible for mass production purposes.

The second most important reason for using intraframe coding is that a situation, or program material, may exist that can make a three-dimensional coding scheme break down and perform poorly, or at least no better than the intraframe coding scheme. For example, 24 frame per second movies can easily fall into this category since the integration time, due to the mechanical shutter, is relatively short. This short integration time allows a higher degree of temporal aliasing than in TV cameras for rapid motion. The assumption of frame to frame correlation breaks down for rapid motion as it becomes jerky. Practical consideration of frame to frame registration error, which is already noticeable on home videos become worse at higher resolution.

An additional reason for using intraframe coding is that a three-dimensional coding scheme is more difficult to standardize when both 50 Hz and 60 Hz power line frequencies are involved. The use of an intraframe scheme, being a digital approach, can adapt to both 50 Hz and 60 Hz operation, or even to 24 frame per second movies by trading off frame rate versus spatial resolution without inducing problems of standards version.

Although the present invention is described primarily with respect to black and white, the overhead for coding color information is surprisingly small, on the order of 10 to 15% of the bits needed for the luminance. Because of the low spatial sensitivity of the eye to color, most researchers have converted a color picture from RGB space to YIQ space, sub-sample the I and Q components by a factor of four in horizontal and vertical direction. The resulting I and Q components are coded similarly

as Y(luminance). This technique requires 6.25% overhead each for I and Q component. In practice, the coded Q component requires even less data than the I component. It is envisioned that no significant loss in color fidelity will result when utilizing this class of color coding techniques.

In the implementation the DCT coding techniques, the blocking effect is the single most important impairment to image quality. However, it has been realized that the blocking effect is reduced when a smaller sized DCT is used. The blocking effect becomes virtually invisible when a 2 x 2 DCT is used. However, when using the small-sized DCT, the bit per pixel performance suffers somewhat. However, a small-sized DCT helps the most around sharp edges that separate relatively blank area. A sharp edge is equivalent to a step signal which has significant components at all frequencies. When quantized, some of the low energy coefficients are truncated to zero. This quantization error spreads over the block. This effect is similar to a two-dimensional equivalent of the Gibbs phenomenon, i.e. the ringing present around a step pulse signal when part of the high frequency components are removed in the reconstruction process. When adjacent blocks do not exhibit similar quantization error, the block with this form of error stands out and creates the blocking effect. Therefore by using smaller DCT block sizes the quantization error becomes confined to the area near the edge since the error cannot propagate outside the block. Thereby, by using the smaller DCT block sizes in the busy areas, such as at edges, the error is confined to the area along the edge. Furthermore, the use of the small DCT block sizes is further enhanced with respect to subjective quality of the image due to the spatial masking phenomena in the eye that hides noise near busy areas.

The adaptive block size DCT technique implemented in the present invention may be simply described as a compare-and-replace scheme. A 16 x 16 pixel data array or block of the image is coded as in the fixed block size DCT techniques, however, block sizes of 16 x 16, 8 x 8, 4 x 4 and 2 x 2 are used. For each 4 x 4 block, the number of bits to code the block by using four 2 x 2 sub-blocks inside the 4 x 4 block is examined. If the sum of the four 2 x 2 sub-blocks is smaller than the bits needed to code it as a 4 x 4 block, the 4 x 4 block is replaced by four 2 x 2 sub-blocks. Next, each of the 8 x 8 blocks are examined to determine if they can in turn be replaced by four 4 x 4 sub-blocks which were optimized in the previous stage.

Similarly, the 16 x 16 block is examined to determine if it can be replaced by four 8 x 8 sub-blocks that were optimized in the previous stage. At each stage the optimum block/sub-block size is chosen so that the resulting block size assignment is optimized for the 16 x 16 block.

Since 8-bits are used to code the DC coefficients regardless of the block size, utilization of small blocks results in a larger bit count. For this reason, 2 x 2 blocks are used only when their use can lower the bit count. The resulting sub-block structure can be conveniently represented by an inverted quadtree (as opposed to a binary tree), where the root corresponding to the 16 x 16 block in each node has four possible branches corresponding to four sub-blocks. An example of a possible inverted quadtree structure is illustrated in Figure 3b.

Each decision to replace a block with smaller sub-blocks requires one bit of information as overhead. This overhead ranges from one bit for a 16 x 16 block up to 21 bits (1+4+16) when 4 x 4 and 2 x 2 sub-blocks are used everywhere within in the 16 x 16 block. This overhead is also incorporated into the decision making process to ensure that the adaptive block size DCT scheme always uses the least number of bits to code each 16 x 16 block.

Although block sizes discussed herein as being NxN in size, it is envisioned that various block sizes may be used. For example an NxM block size may be utilized where both N and M are integers with M being either greater than or lesser than N. Another important aspect is that the block is divisible into at least one level of sub-blocks, such as N/i x N/i, N/i x N/j, N/i x M/j, and etc. where i and j are integers. Furthermore, the exemplary block size as discussed herein is a 16X16 pixel block with corresponding block and sub-blocks of DCT coefficients. It is further envisioned that various other integer such as both even or odd integer values may be used, e.g. 9x9.

Due to the importance of these overhead bits for the quadtree, these bits need to be protected particularly well against channel errors. One can either provide an extra error correction coding for these important bits or provide and error recovery mechanism so that the effect of channel errors is confined to a small area of the picture.

The adaptive block size DCT compression scheme of the present invention can be classified as an intraframe coding technique, where each

frame of the image sequence is encoded independently. Accordingly, a single frame still picture can be encoded just as easily without modification. The input image frame is divided into a number of 16 x 16 pixel data blocks with encoding performed for each block. The main distinction of the compression scheme of the present invention resides in the fact that the 16 x 16 block is adaptively divided into sub-blocks with the resulting sub-blocks at different sizes also encoded using a DCT process. By properly choosing the block sizes based on the local image characteristics, much of the quantization error can be confined to small sub-blocks. Accordingly small sub-blocks naturally line up along the busy area of the image to their perceptual visibility of the noises lower than in blank areas.

In review, a conventional or fixed block size DCT coding assigns a fixed number of bits to each block such that any quantization noise is confined and distributed within the block. When the severity or the characteristics of the noise between adjacent blocks are different, the boundary between the blocks become visible with the effect commonly known as a blocking artifact. Scene adaptive DCT coding assigns a variable number of bits to each block thereby shifting the noise between fixed sized blocks. However, the block size is still large enough, usually 16 x 16, such that some blocks contain both blank and busy parts of the image. Hence the blocking artifact is still visible along image detail such as lines and edges. Using smaller block sizes such as 8 x 8 or 4 x 4 can greatly reduce the blocking artifact, however, at the expense of a higher data rate. As a result the coding efficiency of DCT drops as the block size gets smaller.

The present invention implements an adaptive block size DCT technique which optimally chooses block size such that smaller blocks are used only when they are needed. As a result, the blocking artifact is greatly reduced without increasing the data rate. Although a number of different methods can be devised that determine block size assignment, an exemplary illustration of an embodiment is provided which assigns block sizes such that the total number of bits produced for each block is minimized.

Figures 1 and 2 illustrate an exemplary implementation of the adaptive block size DCT transform image signal compression scheme for converting NxN pixel data blocks into whole bit coded data. As discussed herein for purposes of illustration N=16. Figure 1 illustrates the

implementation of the DCT transform and block size determination elements. Figure 2 illustrates the DCT coefficient data block selection according to the block size determination along with composite DCT coefficient data block bit coding.

In Figure 1, an image signal as represented by a 16 x 16 block of digitized pixel data is received from the frame buffer (not shown). The pixel data may be either 8 bit black and white image data or 24 bit color image data. The 16 x 16 pixel block is input to a 16 x 16 two-dimensional discrete cosine transform (DCT) element 10a. The 16 x 16 pixel block is also input as four 8 x 8 pixel blocks to 8 x 8 DCT element 10b, as eight 4 x 4 pixel blocks to 4 x 4 DCT element 10c., and as sixty-four 2 x 2 pixel blocks to 2x2 DCT element 10d. DCT elements 10a-10d may be constructed in integrated circuit form as is well known in the art.

DCT elements 10a-10d perform two-dimensional DCT operations on each respectively sized input block of pixel data. For example, DCT element 10a performs a single 16 x 16 transform operation, DCT element 10b performs four 8 x 8 DCT operations, DCT element 10c performs sixteen 4 x 4 DCT operations, while DCT element 10d performs sixty-four 2 x 2 DCT operations. Transform coefficients are output from each DCT element 10a-10d to a respective quantizer look up table 12a-12d.

Quantizer lookup tables 12a-12d may be implemented in conventional read only memory (ROM) form with memory locations containing quantization values. The value of each transform coefficient is used to address a corresponding memory location to provide an output data signal indicative of a corresponding quantized transform coefficient value. The output of quantizer lookup table 12a, indicated by the reference signal QC16, is a 16 x 16 block of quantized DCT coefficient values. The output of quantizer lookup table 12b, indicated by the reference signal QC8, is comprised of a data block of four 8 x 8 sub-blocks of quantized DCT coefficient values. The output of quantizer lookup table 12c, indicated by the reference signal QC4, is comprised of a data block of four 4 x 4 sub-blocks of quantized DCT coefficient values. And finally, the output of quantizer lookup table 12d, indicated by the reference signal QC2, is comprised of a data block of sixty-four 2 x 2 sub-blocks of quantized DCT coefficient. Although not illustrated, the DC (lowest frequency)

coefficients of each transform may be optionally treated separately rather than directly through the corresponding quantizer lookup table.

The outputs of quantizer lookup tables 12a-12d are respectively input to code length lookup tables 14a-14d. The quantized DCT coefficient values are each coded using variable length code, such as a Huffman code, in order to minimize the data rate. Code words and corresponding code lengths are found in the form of code length look up tables 14a-14d. Each of the quantized DCT coefficients QC2, QC4, QC8, and QC16 are used to look up in the code length tables the corresponding number of bits required to code each coefficient. Code length lookup tables 14a-14d may be implemented in read only memory form with the DCT coefficients addressing memory locations which contain respective code length values.

The number of bits required to code each block or sub-block is then determined by summing the code lengths in each block and sub-block. The output of 256 code length values from code length lookup table 14a is provided to code length summer 16a which sums all 256 code lengths for the 16 x 16 block. The output from code length summer 16a is the signal CL16, a single value indicative of the number of bits required to code the 16x16 block of quantized DCT coefficients. The 256 code length values output from code length lookup table 14b is provided to code length summer 16b. In code length summer 16b the number of bits required to code each 8 x 8 DCT coefficient sub-block is determined by summing the code length in each 8x8 sub-block. The output of code length summer 16b is four values indicated by the reference signal CL8 with each value being the sum of sixty-four code lengths in each of the four 8 x 8 blocks. Similarly, code length summer 16c is used to sum the code length in each of the 4 x 4 sub-blocks as output from code length lookup table 14c. The output of code length summer is 16c sixteen values indicated by the reference signal CL4 with each value being the sum of the sixteen code lengths in each of the sixteen 4 x 4 sub-blocks. Code length summer 16d is similarly used in determining the number of bits necessary to code each 2 x 2 sub-block as output from code length lookup table 14d. Code length summer 16d provides sixty-four output values indicated by the reference signal CL2 with each value being the sum of the four code lengths in a respective one of the sixty-four 2 x 2 blocks. The values CL8, CL4, and CL2

are also identified with block position orientation indicia for discussion later herein. the position indicia is a simple x-y coordinate system with the position indicated by the subscript (x,y) associated with the values CL8, CL4, and CL2.

The block size assignment (BSA) is determined by examining values of CL2, CL4, CL8 and CL16. Four neighboring entries of $CL2_{(x,y)}$ are added and the sum is compared with the corresponding entry in $CL4_{(x,y)}$. The output of $CL2_{(x,y)}$ from code length summer 16d is input to adder 18 which adds the four neighboring entries and provides a sum value $CL4'_{(x,y)}$. For example, the values representative of blocks $CL2_{(0,0)}$, $CL_{(0,1)}$, $CL2_{(1,0)}$, and $CL2_{(1,1)}$ are added to provide the value $CL4'_{(0,0)}$. The value output from adder 18 is the value $CL4'_{(x,y)}$ which is compared with the value $CL4_{(x,y)}$ output from code length summer 16c. The value $CL4'_{(x,y)}$ is input to comparator 20 along with the value $CL4_{(x,y)}$. Comparator 20 compares the corresponding input values from adder 18 and code length summer 16c so as to provides a bit value, P, that is output to a P register (Figure 2) and as a select input to multiplexer 22. In the example as illustrated in Figure 1, the value $CL4'_{(0,0)}$ is compared with the value $CL4_{(0,0)}$. If the value $CL4_{(x,y)}$ is greater than the summed values of $CL4'_{(x,y)}$, comparator 20 generates a logical one bit, "1", that is entered into the P register. The "1" bit indicates that a corresponding 4 x 4 block of DCT coefficients can be coded more efficiently using four 2 x 2 sub-blocks. If not, a logical zero bit, "0", is entered into the P register, indicating that the 4 x 4 block is coded more efficiently using the corresponding 4 x 4 block.

The output of code length summer 16c and adder 18 are also provided as data inputs to multiplexer 22. In response to the "1" bit value output from comparator 20, multiplexer 22 enables the $CL4'_{(x,y)}$ value to be output therefrom to adder 24. However should the comparison result in a "0" bit value being generated by comparator 20, multiplexer 22 enables the output $CL4_{(x,y)}$ from code length summer 16c to be input to adder 24. Adder 24 is used to sum the data input therefrom, as selected from the comparisons of the values of $CL4_{(x,y)}$ and $CL4'_{(x,y)}$. The result of the sixteen comparisons of the $CL4_{(x,y)}$ and the $CL4'_{(x,y)}$ data is added in adder 24 to generate a corresponding $CL8'_{(x,y)\ value}$. For each of the sixteen comparisons of the $CL4_{(x,y)}$ and $CL4'_{(x,y)\ values}$, the comparison result bit is sent to the P register.

The next stage in the determination of block size assignment is similar to that discussed with respect to the generation and comparison of the values CL4 and CL4'. The output of $CL8'_{(x,y)}$ is provided as an input to comparator 26 along with the output $CL8_{(x,y)}$ from code length summer 16b. If the corresponding entry in $CL8_{(x,y)}$ is greater than the summed value $CL8'_{(x,y)}$, comparator 26 generates a "1" bit which is output to the Q register (Figure 2). The output of comparator 26 is also provided as a selected input to multiplexer 28 which also receives the values $CL8_{(x,y)}$ and $CL8'_{(x,y)}$ respectively from code length summer 16b and adder 24. Should the value output from comparator 26b a "1" bit the $CL8'_{(x,y)}$ value is output from multiplexer 28 to adder 30. However, should the value $CL8'_{(x,y)}$ be greater than the value $CL8_{(x,y)}$, comparator 26 generates a "0" bit that is sent to the Q register and also to the select input of multiplexer 28. Accordingly, the value $CL8_{(x,y)}$ is then input to adder 30 via multiplexer 28. Comparison results of comparator 26 are the Q values sent to the Q register. Again, a "1" bit indicates that for the corresponding 8 x 8 block of DCT coefficients may be more efficiently coded by smaller blocks such as all 4 x 4 blocks, all 2 x 2 blocks or a combination thereof as optimally determined by the smaller block comparisons. A "0" bit indicates that the corresponding 8 x 8 block of DCT coefficients can be more efficiently coded than any combination of smaller blocks.

The summed values input to adder 30 are provided as an output value CL16' for input to comparator 32. A second input is provided to comparator 32 as the value CL16 output from by code length summer 16a. Comparator 32 preforms a single comparison of the value CL16 and CL'16. Should the value CL16 be greater than the value CL16' a "1" bit is entered into the R register (Figure 3) . A "1" bit input to the R register is indicative that the block may be coded more efficiently using sub-blocks rather than a single 16 x 16 block. However should the value CL16' be greater than the value CL16, comparator 32 outputs a "0" bit to the R register. The "0" bit in the R register is indicative that the block of DCT coefficients may be coded more efficiently as a 16 x 16 block.

Comparator 32 is also provides the output R bit as a select input to multiplexer 34. Multiplexer 34 also has inputs for receiving the CL16 and CL16' values respectively provided from code length summer 16a and adder 30. The output from multiplexer 34 is the value CL16 should the

R output bit be a "0" while the value CL16' is output should the R output bit be a "1". The output of multiplexer 34 is a value indicative of the total bits to be transmitted.

It should be noted that the overhead bits, vary from one bit to up to twenty-one bits (1+4+16) when 4 x 4 and 2 x 2 blocks are used everywhere within the 16 x 16 block.

In Figure 2, the P value output from comparator 20 (Figure 1) is input serially to a sixteen-bit register, P register 40. Similarly, the output from comparator 26 is input serially to a four-bit register, Q register 42. Finally, the output from comparator 32 is input serially to a one-bit register, R register 44. The output from P register 40 is provided as a P output to the select input of multiplexer 46. Multiplexer 46 also has inputs as the QC2 and QC4 values respectively output from Quantizer lookup tables 12b and 12c. The output of multiplexer 46 is provided as an input to multiplexer 48, which also has as a second input for the QC8 values as output from quantizer lookup table 12b. A select input to multiplexer 48 is provided from the output of Q register 42. The output of multiplexer 48 is coupled as one input to multiplexer 50. The other input of multiplexer 50 is coupled to the output of quantizer lookup table 12a for receiving the values QC16. The select input of multiplexer 50 is coupled to the output of R register 44 so as to receive the output bit R.

As illustrated in Figure 2, P register 40 includes a sequence of bit positions, 0-15, with corresponding bit values as determined by the comparison process as discussed with reference to Figure 1. Similarly Q register 42 and R register 44 respectively have bit position 0-3 and 0 with corresponding data as determined with reference to Figure 1. The data in the P, Q and R registers as illustrated in Figure 2 is merely for the purpose of illustration.

As illustrated in Figure 2, the value of P register 40 bit is used to select via multiplexer 4b, QC2 data (four 2 x 2 blocks of quantized transform coefficients) or the corresponding QC4 data (a 4 x 4 block of quantized transform coefficients). Multiplexer 48, in response to the value of the bit output from Q register 42 selects between the output of multiplexer 46 and the value QC8 data. When the Q register bit value is a "1" bit, the output of multiplexer 46 as input to multiplexer 48 is selected for output of multiplexer 48. When the Q register bit value is a "0" bit, the

output of multiplexer 48 is the QC8 value. Therefore, the output bit value of Q register 42 is used to select between four QC4 blocks or sub-blocks of QC2 values as output from multiplexer 46 or a corresponding single 8 x 8 block. As illustrated in Figure 2, the four upper left hand blocks as output from multiplexer 46 include two 2 x 2 blocks with three neighboring 4 x 4 blocks. However with the bit of the Q register being a "0" bit, multiplexer 48 selects the 8 x 8 block as an output. This example illustrates the conditional replacement scheme.

The output of multiplexer 48 is coupled as an input to multiplexer 50. The other input of multiplexer 50 is provided with the Q16 data, the 16 x 16 block of quantized DCT coefficients as provided from quantizer lookup table 12a. The select input to multiplexer 50 is the output bit of the R register. In the example illustrated in Figure 2, the bit output from R register 44 is a "1" bit thus selecting data as output from multiplexer 50 that which was provided from multiplexer 48. Should the R register 44 output bit value be a "0" bit, multiplexer 50 would output the QC16 data.

The multiplexing scheme as illustrated in Figure 2 utilizes the block assignments to multiplex coefficient sub-blocks QC2, QC4, QC8, QC16 values into a composite block of DCT coefficients QC. In essence this step is accomplished by three stages. The first stage conditionally replaces a 4 x 4 block of QC4 with four 2 x 2 sub-blocks according to the content of the P register. The second stage conditionally replaces an 8 x 8 block of QC8 by four 4 x 4 sub-blocks as resulting from the previous stage. The third stage replaces the 16 x 16 block of QC16 by the result of the previous stages if the R register contains a "1" bit.

Figures 3a and 3b respectively illustrate register and BSA bit pattern and corresponding inverted quadtree corresponding thereto. The level of hierarchy involved is that should the bit stored in the R register be a "1", a condition exists which is indicative that the image block may be more efficiently coded using smaller blocks. Similarly, should the Q register contain any "1" bits it further indicates that an 8 x 8 block may be more efficiently coded by smaller blocks. Similarly, should the P register contain any "1" bits it further indicates that a 4 x 4 block may be more efficiently coded using four 2 x 2 blocks. Should any of the registers contain a "0" bit, this indicates that the block or sub-block may be coded more efficiently by using the size block related thereto. For example, even though the value

of the bit in the P register bit 0 position, a "1" bit, indicates that this 4 x 4 block may be more efficiently coded using four 2 x 2 blocks, the bit value Q register indicates that the four 4 x 4 blocks may be more efficiently coded by a single 8 x 8 block. Therefore, the Q register data would override the P register data. Once the P register data was overridden by the P register 0 position bit, the P register bits position 0 - 3, need not be transmitted as part of the block size assignment (BSA) data. However, should a bit position in a higher register be a "1" bit, such as bit position "1" of the Q register, the corresponding P register bits are provided as part of the block size assignment data. As illustrated in Figure 3a, the Q register bit "1" position is a "1" bit and therefore the corresponding P register bits 4 - 7 are provided in the BSA data. On a higher level, since the R register bit is a "1" bit each of the Q register bits are provide in the BSA data.

Returning to Figure 2, the composite block QC contains many zero coefficient values which can be more efficiently coded by run-length codes. The number of consecutive zeros or runs are sent instead of the code words for each zero. In order to maximize the efficiency of the run-length coding, the coefficients are ordered in a predetermined manner such that the occurrence of short runs is minimized. Minimization is done by encoding the coefficients which are likely to be non-zeros first, and then encoding the coefficients that are more likely to be zeros last. Because of the energy compaction property of DCT towards low frequency, and because diagonal details occur less often than horizontal or vertical details, diagonal scan or zig-zag scan of the coefficients is preferred. However, because of the variable block sizes used, the zig-zag scan has to be modified to pick out the low frequency components from each sub-block first, but at the same time follow the diagonal scanning for coefficients of similar frequency, technically when the sum of the two frequency indices are the same.

Accordingly, the output composite block QC from multiplexer 50 is input to zig-zag scan serializer 52 along with the BSA data (P,Q and R). Figure 4a illustrates the zig-zag ordering of the block data within blocks and corresponding sub-blocks. Figure 4b illustrates the ordering in the serialization between blocks and sub-blocks as determined by the BSA data.

The output of zig-zag scan serializer 52, comprised of the ordered 256 quantized DCT coefficients of the composite block QC, is input to

coefficient buffer 54 where they are stored for run-length coding. The serialized coefficients are output from coefficient buffer 54 to run-length coder 56 where run-length coding is preformed to separate out the zeros from the non-zero coefficients. Run-length as well as the non-zero coefficient values are separately provided corresponding to lookup tables. The run-length values are output from run-length coder 56 as an input of run-length code lookup table 58 where the values are Huffman coded. Similarly, the non-zero coefficient values are output from run-length coder 56 as an input to non-zero code lookup table 60 where the values are also Huffman coded. Although not illustrated it is further envisioned that run-length and non-zero code look up tables may be provided for each block size.

The Huffman run-length coded values along with the Huffman non-zero coded values are respectively output from run-length lookup code table 58 and non-zero code lookup table 60 as inputs to bit field assembler 62. An additional input to bit field assembler 62 is the BSA data from the P, Q and R registers. Bit field assembler 62 disregards the unnecessary bits provided from the P, Q and R registers. Bit field assembler 62 assembles the input data with BSA data followed by the combined RL codes and NZ codes. The combined data is output from bit field assembler 62 to a transmit buffer 64 which temporarily stores the data for transfer to the transmitter (not shown).

Figures 5a-5d illustrates an alternate scan and serialization format for the zig-zag scan serializer 52. In Figures 5a-5d, the quantized DCT coefficients are mapped into a one-dimensional string by ordering from low frequency to high frequency. However in the scheme illustrated in Figures 5a-5d the lower order frequencies are taken from each block prior to taking the next higher frequencies in the block. Should all coefficients in a block be ordered, during the previous scan, the block is skipped with priority given to the next block in the scan pattern. Block to block scanning, a was done with the scanning of Figures 4a-4c is a left-to-right, up to down scan priority.

Figure 6 illustrates the implementation of a receiver for decoding the compressed image signal generated according to the parameters of Figures 1 and 2. In Figure 6, the coded word is output from the receiver (not shown) to a receive buffer 100. Receive buffer 100 provides an output

of the code word to BSA separator 102. Received code words include by there nature the BSA, RL codes and NZ codes. All received code words obey the prefix conditions such that the length of each code word need not be known to separate and decode the code words.

BSA separator 102 separates the BSA codes from the RL and NZ codes since the BSA codes are transmitted and received first before the RL and NZ codes. The first received bit is loaded into an internal R register (not shown) similar to that of Figure 2. An examination of the R register determines that if the bit is a "0", the BSA code is only one bit long. BSA Separator 102 also includes Q and P registers that are initially filled with zeros. If the R register contains a "1" bit, four more bits are taken from the receive buffer and loaded into the Q register. Now for every "1" bit in the Q register, four more bits are taken from the receive buffer and loaded into the P register. For every "0" in the Q register, nothing is taken from the receive buffer but four "0" are loaded into the P register. Therefore, the possible lengths of the BSA code is 1, 5, 9, 13, 17 and 21 bits. The decoded BSA data is output from BSA separator 102.

BSA separator 102 further separates, and outputs, the RL codes and NZ codes respectively to RL decode lookup table 104 and NZ decode lookup table 106. Lookup tables 104 and 106 are essentially inverse lookup tables with respect to lookup tables 58 and 60 of Figure 2. The output of lookup table 104 is a value corresponding to the run-length and is input to run-length decoder 108. Similarly the non-zero coefficient values output from lookup table 106 is also input to run-length decoder 108. Run-length decoder 108 inserts the zeros into the decoded coefficients and provides an output to coefficient buffer 110 which temporarily stores the coefficients. The stored coefficients are output to an inverse zig-zag scan serializer 112 which orders the coefficients according to the scan scheme employed. Inverse zig-zag scan serializer 112 receives the BSA signal from separator 102 to assist in proper ordering of the block and sub-block coefficients into a composite coefficient block. The block of coefficient data is output from inverse zig-zag scan serializer 112 and respectively applied to a corresponding inverse quantizer lookup table 114a - 114d. An inverse quantizer value is applied to each coefficient to undo the quantization. Inverse quantizer lookup tables 114a - 114d may be employed as ROM devices which contain the quantization factors from

that of quantizer lookup tables 12a - 12d. The coefficients are output from each of inverse quantizer look up tables 114a - 114d to corresponding inverse discrete cosine transform (IDCT) elements 116a - 116d.

IDCT element 116a forms from the 16 x 16 the IDCT coefficient block, if present, a 16 x 16 pixel data block which is then output to sub-block combiner 118. Similarly, DCT 116b transforms respective 8 x 8 blocks of coefficients, if present, to 8 x 8 blocks of pixel data. The output of IDCT element 116b is provided to sub-block combiner 118. IDCT elements 116c and 116d respective transform the 4 x 4 and 2 x 2 coefficient blocks, if present, to corresponding pixel data blocks which are provided to sub-block combiner 118. Sub-block combiner 118 in addition to receiving the outputs from IDCT elements 116a - 116d also receives the BSA data from separator 102 so as to reconstruct the blocks of pixel data a single 16 x 16 pixel block. The reconstructed 16 x 16 pixel block is output to a reconstruction buffer (not shown).

Figure 7 illustrates in block diagram form a flow chart for signal compression of the present invention. Figure 7 briefly illustrates the steps involved in the processing as discussed with reference to Figure 1. Similarly Figure 8 illustrates the decompression process of transmitted compressed image data to result in the output pixel data. The steps illustrated in Figure 8 are previously discussed with reference to Figure 6.

The present invention utilizes a unique adaptive block size processing scheme which provides substantially improved image quality without making a great sacrifice in the bit per pixel ratio. It is also believed that a bit per pixel ratio of about "1" and even substantially less than this level would provide substantial improvement in image quality sufficient for HDTV applications when using the techniques disclosed herein. It is envisioned that many variations to the invention may be readily made upon review of the present disclosure.

The present invention also envisions the implementation of a new and previously undisclosed transform identified herein as the differential quadtree transform (DQT). The basis for this transform is the recursive application of the 2 x 2 DCT on a quadtree representation of the sub-blocks. At the bottom of the inverted quadtree, for example that illustrated in Figure 3b, the 2 x 2 DCT operation is performed and the node is assigned the DC value of the 2 x 2 DCT transform. The nearest nodes are gathered

and another 2 x 2 DCT is performed. The process is repeated until a DC value is assigned to the root. Only the DC value at the root is coded at a fixed number of bits, typically 8-bits, while the rest are Huffman coded. Because each 2 x 2 DCT operation is nothing more than a sum and a difference of numbers, no multiplications are required, and all coefficients in the quadtree except the DC represent differences of two sums, hence the name DQT. Theoretically this type of transform cannot exceed the performance of 16 x 16 DCT coding. However the implementation of the DQT transform has the advantage of requiring seemingly simple hardware in addition to naturally implementing the adaptive block size coding. Furthermore the quadtree structure allows the coding of the zero coefficients by simply indicating the absence of a subtree when all sub-blocks under the subtree contain only zeros.

The previous description of the preferred embodiments are provided to enable any person skilled in the art to make or use the present invention. The various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of the inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

I claim:

CLAIMS

1. In an image compression system, a sub-system for generating from a block of input pixel data a corresponding composite block of discrete cosine transform (DCT) data, comprising:

transform means for, receiving an input block of pixel data, performing a discrete cosine transform (DCT) operation on said block of pixel data and on at least one predetermined level of constituent sub-blocks thereof, and providing corresponding block and sub-blocks of DCT coefficient values;

block size assignment means for, receiving said block and each sub-block of DCT coefficient values, determining for each block and each corresponding group of sub-blocks of DCT coefficient values a bit count value corresponding to a number of bits required to respectively encode said block and each corresponding group of sub-blocks of DCT coefficient values according to a predetermined coding format, determining from said bit count values ones of said block and corresponding group of sub-blocks of DCT coefficient values requiring a lesser number of bits to encode according to said coding format, and providing a corresponding selection value; and

selection means for, receiving said selection value and said block and each sub-block of DCT coefficient values, selecting ones of block and sub-blocks of DCT coefficients corresponding to said selection value, and providing an output of a corresponding composite block of DCT coefficient values formed from said selected ones of block and sub-blocks of DCT coefficient values.

2. The sub-system of Claim 1 further comprising quantization means coupled between said transform means, said block size assignment means, and said selection means for, receiving said block and sub-blocks of DCT coefficient values, quantizing each DCT coefficient value, and providing each quantized DCT coefficient value to said block size assignment means and said selection means in place of a corresponding DCT coefficient.

3. An image signal compression system comprising:

transform means for, receiving an input block of pixel data, performing a discrete cosine transform (DCT) operation on said block of pixel data and on at least one predetermined level of constituent sub-blocks thereof, and providing corresponding block and sub-blocks of DCT coefficient values;

block size assignment means for, receiving said block and each sub-block of DCT coefficient values, determining for said block and each corresponding group of sub-blocks of DCT coefficient values a bit count value corresponding to a number of bits required to respectively encode said block and corresponding group of sub-blocks of DCT coefficient values according to a predetermined coding format, determining from said bit count values ones of said block and corresponding group of sub-blocks of DCT coefficient values requiring a lesser number of bits to encode according to said coding format, and providing a corresponding selection value;

selection means for, receiving said selection value and said block and each sub-block of DCT coefficient values, selecting ones of block and sub-blocks of DCT coefficients corresponding to said selection value, and providing an output of a corresponding composite block of DCT coefficient values formed from said selected ones of block and sub-blocks of DCT coefficient values;

ordering means for, receiving said composite block of DCT coefficient values, ordering said composite block DCT coefficient values according to a predetermined ordering format, and providing a corresponding output of ordered DCT coefficient values;

encoder means for, receiving said ordered DCT coefficient values, coding said ordered DCT coefficient values according to said predetermined coding format, and providing corresponding coded values; and

assembler means for, receiving said selection value and said coded values, combining said selection value and said coded value as a coded image value representative of said input block of pixel data with said coded image value of a reduced bit count with respect to a bit count of said

input block of pixel data,and providing an output of said coded image value.

4. The system of Claim 3 further comprising:

run-length coder means for, receiving said ordered DCT coefficient values, run-length coding said ordered coefficient values, and providing a corresponding output of non-zero coefficient values and run-length values to said encoder means for coding according to said predetermined coding format.

5. A method for compressing an image signal corresponding to a block of pixel data, comprising the steps of:

providing a block of pixel data capable of division into varying levels of constituent sub-blocks;

performing a discrete cosine transform (DCT) operation on said block of pixel data so as to provide a corresponding block of DCT coefficients;

performing a DCT operation on each sub-block of pixel data in at least one level of pixel data sub-blocks so as to provide corresponding sub-blocks of DCT coefficients;

determining for each block and corresponding group of sub-blocks of DCT coefficients a bit count value corresponding to a number of bits necessary to encode each block and group of sub-blocks according to a predetermined coding format;

determining, from said bit count values, ones of said block and corresponding groups of sub-blocks of DCT coefficients corresponding to a lower number of bits necessary to encode each block and corresponding group of sub-blocks of DCT coefficients so as to provide a corresponding block size assignment value;

assembling, according to said block assignment value, a composite block of DCT coefficients from said ones of said block and corresponding groups of sub-blocks of DCT coefficients having a lower bit encoding number;

IPEA/US 12 MAY 199?
PCT/US 91/01350

ordering said composite block of DCT coefficients according to a predetermined ordering format so as to provide a serial string of ordered DCT coefficients;

encoding said string of ordered DCT coefficients according to said predetermined coding format so as to provide a set of coded DCT coefficients; and

assembling said block assignment value and said coded DCT coefficients into coded image data representative of said block of pixel data, said coded image data being of a reduced bit count with respect to a bit count of said block of pixel data.

6. An image compression system for compressing a 16x16 block of pixel data comprising:

first transform means for receiving and performing 2x2 discrete cosine transform (DCT) operations on said block of pixel data, and providing a corresponding block of sixty-four 2x2 DCT coefficient value sub-blocks;

second transform means for receiving and performing 4x4 DCT operations on said block of pixel data, and providing a corresponding block of sixteen 4x4 DCT coefficient value sub-blocks;

third transform means for receiving and performing 8x8 DCT operations on said block of pixel data, and providing a corresponding block of four 8x8 DCT coefficient value sub-blocks;

fourth transform means for receiving and performing a 16x16 DCT operation on said block of pixel data, and providing a corresponding 16x16 DCT coefficient value block;

first memory means for storing bit values each corresponding to a number of bits required to encode according to a predetermined encoding format each different 2x2 DCT coefficient value, said first memory means for receiving each 2x2 DCT coefficient value sub-block and providing a corresponding first set of bit values;

second memory means for storing bit values each corresponding to a number of bits required to encode according to a predetermined encoding format each different 4x4 DCT coefficient value, said second

SUBSTITUTE SHEET
IPEA/US

memory means for receiving each 4x4 DCT coefficient value sub-block and providing a corresponding second set of bit values;

third memory means for storing bit values each corresponding to a number of bits required to encode according to a predetermined encoding format each different 8x8 DCT coefficient value, said third memory means for receiving each 8x8 DCT coefficient value sub-blocks and providing a corresponding third set of bit values;

fourth memory means for storing bit values each corresponding to a number of bits required to encode according to a predetermined encoding format each different 16x16 DCT coefficient value, said fourth memory means for receiving said 16x16 DCT coefficient value block and providing a corresponding fourth set of bit values;

first code length summer means for, receiving said first set of bit values, summing ones of said first set of bit values corresponding to each respective 2x2 DCT coefficient value sub-block and providing corresponding first code length sum values;

second code length summer means for, receiving said second set of bit values, summing ones of said second set of bit values corresponding to each respective 4x4 DCT coefficient value sub-block and providing corresponding second code length sum values;

third code length summer means for, receiving said third set of bit values, summing ones of said third set of bit values corresponding to each respective 8x8 DCT coefficient value sub-block and providing corresponding third code length sum values;

fourth code length summer means for, receiving said fourth set of bit values, summing said fourth set of bit values and providing a corresponding fourth code length sum value;

first adder means for receiving and adding predetermined groups of said first code length sum values, and providing corresponding first adder values;

first comparator means for receiving and comparing each first adder value with a predetermined corresponding second code length sum value, and providing a corresponding first comparison value;

first multiplexer means for receiving each first adder value, each corresponding second code length sum value, and each corresponding first

IPEA/US 12 MAY 1992
PCT/US 91/01350

comparison value and for providing a first multiplexer value selected from one of each first adder value and corresponding second code length sum value in accordance with said first comparison value;

first register means for receiving and storing in a predetermined order each first comparison value;

second adder means for receiving and adding predetermined groups of first multiplexer values, and providing corresponding second adder values;

second comparator means for receiving and comparing each second adder value with a predetermined corresponding third code length sum value, and providing a corresponding second comparison value;

second multiplexer means for receiving each second adder value, each corresponding third code length sum value, and each corresponding second comparison value and for providing a second multiplexer value selected from one of each second adder value and corresponding third code length sum value in accordance with said second comparison value;

second register means for receiving and storing in a predetermined order each second comparison value;

third adder means for receiving and adding said second multiplexer values, and providing a third adder value;

third comparator means for receiving and comparing said third adder value with said fourth code length sum value, and providing a corresponding third comparison value;

third register means for receiving and storing said third comparison value;

third multiplexer means for receiving each of said 2x2 DCT coefficient value sub-blocks, said 4x4 DCT coefficient value sub-blocks and a corresponding predetermined one of said first register means stored first comparison values, and for providing a third multiplexer value selected from one of groups of said 2x2 DCT coefficient value sub-blocks and a corresponding one of said 4x4 DCT coefficient value sub-blocks in accordance with said stored corresponding first comparison value;

fourth multiplexer means for receiving each third multiplexer value, said 8x8 DCT coefficient value sub-blocks and a corresponding predetermined one of said second register means stored second

comparison values, and for providing a fourth multiplexer value selected from one each third multiplexer value and a corresponding one of said 8x8 DCT coefficient value sub-blocks in accordance with said stored corresponding second comparison value;

fifth multiplexer means for receiving each fourth multiplexer value, said 16x16 DCT coefficient value block and a corresponding predetermined one of said third register means stored third comparison values, and for providing a fifth multiplexer value selected from one each fourth multiplexer value and said 16x16 DCT coefficient value block in accordance with said stored corresponding third comparison value;

serializer means for receiving said fifth multiplexer value and each of said stored first, second and third comparison values, for ordering DCT coefficient values of said fifth multiplexer value in accordance with said stored first, second and third comparison values and for providing a corresponding output of serialized DCT coefficient values;

encoder means for receiving and encoding said serialized DCT coefficients according to said predetermined encoding format and for providing corresponding encoder output values; and

assembler means for receiving and ordering said encoder output values and said stored first, second and third comparison values according to a predetermined format, and for providing a corresponding assembler output values.

7. The system of Claim 6 further comprising:

first quantizer means for receiving and quantizing said block of 2x2 DCT coefficient values according to a first predetermined quantization scheme, and for providing a corresponding block of 2x2 quantized DCT coefficient values to said first memory means;

second quantizer means for receiving and quantizing said block of 4x4 DCT coefficient values according to a second predetermined quantization scheme, and for providing a corresponding block of 4x4 quantized DCT coefficient values to said second memory means;

third quantizer means for receiving and quantizing said block of 8x8 DCT coefficient values according to a third predetermined quantization

IPEA/US 12 MAY 1992
PCT/US 91/01350

scheme, and for providing a corresponding block of 8x8 quantized DCT coefficient values to said third memory means; and

fourth quantizer means for receiving and quantizing said block of 16x16 DCT coefficient values according to a fourth predetermined quantization scheme, and for providing a corresponding block of 16x16 quantized DCT coefficient values to said fourth memory means.

8. The system of Claim 6 wherein each of said first, second, third and fourth transform means comprises a DCT transform processor.

9. The system of Claim 6 further comprising sixth multiplexer means for receiving said third adder value, said fourth code length sum value, and each corresponding third comparison value and for providing a third multiplexer value selected from said third adder value and said fourth code length sum value in accordance with said third comparison value, said third multiplexer value indicative of a total number of bits of encoded DCT coefficients for transmission.

10. The system of Claim 6 further comprising coefficient buffer means for receiving and storing said serialized DCT coefficient values, and for providing said stored serialized DCT coefficient values to said encoder means.

11. The system of Claim 6 wherein said encoder means comprises:

run-length encoder means for receiving and run-length encoding said serialized DCT coefficient values, and for providing a run-length value and and a non-zero coefficient value;

run-length memory means for storing predetermined code values each corresponding to a different run-length value, said run-length memory means for receiving each of said run-length values and providing a corresponding coded run-length value; and

non-zero coefficient memory means for storing predetermined code values each corresponding to a different non-zero coefficient value, said non-zero coefficient memory means for receiving each of said non-zero

IPEA/US 12 MAY 1992
PCT/US 91/01350

coefficient values and providing a corresponding coded non-zero coefficient.

12. The system of Claim 6 wherein:

said first comparison value is indicative of one of each group of said 2x2 DCT coefficient value sub-blocks and corresponding 4x4 DCT coefficient value sub-block requiring a lesser of number of bits to encode according to said predetermined encoding format;

said second comparison value is indicative of one of each group of said 4x4 DCT coefficient value sub-blocks and corresponding 8x8 DCT coefficient value sub-block requiring a lesser of number of bits to encode according to said predetermined encoding format; and

said third comparison value is indicative of a group of said 8x8 DCT coefficient value sub-blocks and said 16x16 DCT coefficient value block requiring a lesser of number of bits to encode according to said predetermined encoding format.

13. An image signal compression system comprising:

transform means for, receiving a 16x16 block of pixel data, performing a 16x16 discrete cosine transform (DCT) operation on said block of pixel data and providing a corresponding 16x16 block of DCT coefficient values, performing four 8x8 DCT operations on said block of pixel data and providing a corresponding block of four 8x8 DCT coefficient value sub-blocks, performing sixteen 4x4 DCT operations on said block of pixel data and providing a corresponding block of sixteen 4x4 DCT coefficient value sub-blocks, performing sixty-four 2x2 DCT operations on said block of pixel data and providing a corresponding block of sixty-four 2x2 DCT coefficient value sub-blocks;

quantization means for receiving and quantizing said 16x16 block of DCT coefficient values and each 8x8, 4x4 and 2x2 sub-block of DCT coefficient values, and for providing a corresponding 16x16 block of quantized DCT coefficients and corresponding 8x8, 4x4 and 2x2 sub-block of quantized DCT coefficient values;

block size assignment means for, receiving said 16x16 block of quantized DCT coefficient values and each 8x8, 4x4 and 2x2 sub-block of

quantized DCT coefficient values, determining for said 16x16 block of quantized DCT coefficient values and each of corresponding groups of 8x8, 4x4 and 2x2 sub-blocks of quantized DCT coefficient values a bit count value corresponding to a number of bits required to respectively encode said 16x16 block of quantized DCT coefficient values and said corresponding groups of 8x8, 4x4 and 2x2 sub-blocks of quantized DCT coefficient values according to a predetermined coding format, determining from said bit count values ones of said 16x16 block of DCT coefficient values and said corresponding groups of 8x8, 4x4 and 2x2 sub-blocks of quantized DCT coefficient values requiring a lesser number of bits to encode according to said coding format, and providing a corresponding selection value;

selection means for, receiving said selection value and said 16x16 block of quantized DCT coefficient values and each 8x8, 4x4 and 2x2 sub-block of quantized DCT coefficient values, selecting ones of said 16x16 block of quantized DCT coefficient values and groups of 8x8, 4x4 and 2x2 sub-block of quantized DCT coefficient values in accordance with said selection value, and providing an output of a corresponding composite block of quantized DCT coefficient values formed from said selected ones of said 16x16 block of quantized DCT coefficient values and groups of 8x8, 4x4 and 2x2 sub-block of quantized DCT coefficient values;

ordering means for, receiving said composite block of quantized DCT coefficient values, ordering said composite block of quantized DCT coefficient values according to a predetermined ordering format, and providing a corresponding output of ordered quantized DCT coefficient values;

encoder means for, receiving said ordered quantized DCT coefficient values, coding said ordered quantized DCT coefficient values according to said predetermined coding format, and providing corresponding coded values; and

assembler means for, receiving said selection value and said coded values, combining said selection value and said coded value as a coded image value representative of said block of pixel data with said coded image value of a reduced bit count with respect to a bit count of said block of pixel data, and providing an output of said coded image value.

IPEA/US 12 MAY 1992
PCT/US 91/01350

14. The system of Claim 13 wherein said transform means comprises:

a 2x2 DCT element having an input for receiving said block of pixel data, and an output coupled to said quantization means;

a 4x4 DCT element having an input for receiving said block of pixel data, and an output coupled to said quantization means;

an 8x8 DCT element having an input for receiving said block of pixel data, and an output coupled to said quantization means; and

a 16x16 DCT element having an input for receiving said block of pixel data, and an output coupled to said quantization means.

15. The system of Claim 13 wherein said quantization means comprises:

first memory means for, storing first quantization values, receiving each DCT coefficient value of each 2x2 sub-block of DCT coefficients, and providing corresponding quantized DCT coefficient values;

second memory means for, storing second quantization values, receiving each DCT coefficient value of each 4x4 sub-block of DCT coefficient values, and providing corresponding quantized DCT coefficient values; and

third memory means for, storing third quantization values, receiving each DCT coefficient value of each 8x8 sub-block of DCT coefficient values, and providing corresponding quantized DCT coefficient values; and

fourth memory means for, storing fourth quantization values, receiving each DCT coefficient value of said 16x16 block of DCT coefficients value, and providing corresponding quantized DCT coefficient values.

16. The system of Claim 14 wherein said quantization means comprises:

a first quantization memory having an address input coupled to said 16x16 DCT element output, and an output coupled to said block size assignment means and said selection means;

a second quantization memory having an address input coupled to said 8x8 DCT element output, and an output coupled to said block size assignment means and said selection means;

a third quantization memory having an address input coupled to said 4x4 DCT element output, and an output coupled to said block size assignment means and said selection means; and

a fourth quantization memory having an address input coupled to said 2x2 DCT element output, and an output coupled to said block size assignment means and said selection means.

17. The system of Claim 16 wherein said block size assignment means comprises:

a first code length memory having an address input coupled to said first quantizer memory output, and an output;

a second code length memory having an address input coupled to said second quantizer memory output, and an output;

a third code length memory having an address input coupled to said third quantizer memory output, and an output;

a fourth code length memory having an address input coupled to said fourth quantizer memory output, and an output;

a first code length summer having an input coupled to said first code length memory output, and an output;

a second code length summer having an input coupled to said second code length memory output, and an output;

a third code length summer having an input coupled to said third code length memory output, and an output;

a fourth code length summer having an input coupled to said fourth code length memory output, and an output;

a first adder having an input coupled to said first code length summer output, and an output;

a first comparator having a pair of inputs respectively coupled to said second code length summer output and said first adder output, and an output;

a first multiplexer having a pair of data inputs respectively coupled to said second code length summer output and said first adder output, a control input coupled said first comparator output, and an output;

a second adder having an input coupled to said first multiplexer output, and an output;

a second comparator having a pair of inputs respectively coupled to said third code length summer output and said second adder output, and an output;

a second multiplexer having a pair of data inputs respectively coupled to said third code length summer output and said second adder output, a control input coupled said second comparator output, and an output;

a third adder having an input coupled to said second multiplexer output, and an output; and

a third comparator having a pair of inputs respectively coupled to said fourth code length summer output and said third adder output, and an output.

18. The system of Claim 17 wherein said block size assignment means further comprises a third multiplexer having a pair of data inputs respectively coupled to said fourth code length summer output and said third adder output, a control input coupled said third comparator output, and an output.

19. The system of Claim 17 wherein said selection means comprises:

a first register having an input coupled to said first comparator output, and an output;

a second register having an input coupled to said second comparator output, and an output;

a third register having an input coupled to said third comparator output, and an output

a third multiplexer having a pair of data inputs respectively coupled to said first and second quantization memory outputs, a control input coupled said first register output, and an output;

IPEA/US 12 MAY 1992
PCT/US 91/01350

a fourth multiplexer having a pair of data inputs respectively coupled to said third quantization memory output and said third multiplexer output, a control input coupled said second register output, and an output; and

a fifth multiplexer having a pair of data inputs respectively coupled to said fourth quantization memory output and said fourth multiplexer output, a control input coupled said second register output, and an output.

20. The system of Claim 19 wherein said ordering means comprises a zig-zag scan serializer having a coefficient input and a control input coupled to said first, second and third register outputs.

21. The system of Claim 20 further comprising:

a coefficient buffer having an input coupled to said zig-zag scan serializer output, and an output; and

a run-length coder having an input coupled to said coefficient buffer, and a non-zero coefficient value output and a run-length value output coupled to said encoder means.

22. The system of Claim 13 wherein said encoder means comprises:

a Huffman coder having an input coupled to said ordering means and an output coupled to said assembler means.

23. The system of Claim 21 wherein said encoder means comprises a Huffman coder comprised of:

a first encoder memory, storing Huffman code values, having an input coupled to said non-zero coefficient value output and an output coupled to said assembler means; and

a second encoder memory, storing Huffman code values, having an input coupled to said run-length coefficient value output and an output coupled to said assembler means.

IPEA/US 12 MAY 199?
PCT/US 91/01350

24. A method for image signal compression of a block of pixel data comprising the steps of:

performing a 16x16 discrete cosine transform (DCT) operation on an 16x16 block of pixel data so as to provide a corresponding 16x16 block of DCT coefficient values;

performing four 8x8 DCT operations on said block of pixel data so as to provide a corresponding block of four 8x8 DCT coefficient value sub-blocks;

performing sixteen 4x4 DCT operations on said block of pixel data so as to provide a corresponding block of sixteen 4x4 DCT coefficient value sub-blocks;

performing sixty-four 2x2 DCT operations on said block of pixel data so as to provide a corresponding block of sixty-four 2x2 DCT coefficient value sub-blocks;

determining bit count values corresponding to a number of bits required to respectively encode said 16x16 block of DCT coefficient values and predetermined groups of 8x8, 4x4 and 2x2 sub-blocks of DCT coefficient values according to a predetermined coding format;

determining, from said bit count values, ones of said 16x16 block of DCT coefficient values and said corresponding groups of 8x8, 4x4 and 2x2 sub-blocks of DCT coefficient values requiring a lesser number of bits to encode according to said predetermined coding format so as to provide a corresponding selection value;

selecting ones of said 16x16 block of DCT coefficient values and groups of 8x8, 4x4 and 2x2 sub-block of DCT coefficient values in accordance with said selection value;

generating a composite block of DCT coefficient values formed from said selected ones of said 16x16 block of DCT coefficient values and groups of 8x8, 4x4 and 2x2 sub-block of DCT coefficient values;

ordering said composite block of DCT coefficient values according to a predetermined ordering format so as to provide a corresponding output of ordered DCT coefficient values;

encoding said ordered DCT coefficient values according to said predetermined coding format so as to provide corresponding coded values; and

assembling said selection value and said coded values in a predetermined order as a coded image value representative of said block of pixel data.

25. The method of Claim 24 further comprising the steps of:

quantizing each DCT coefficient value of said 16x16 block of DCT coefficient values so as to provide to a corresponding 16x16 block of quantized DCT coefficient values;

quantizing each DCT coefficient value of each 8x8 sub-block of DCT coefficient values so as to provide corresponding 8x8 sub-blocks of quantized DCT coefficient values;

quantizing each DCT coefficient value of each 4x4 sub-block of DCT coefficient values so as to provide corresponding 4x4 sub-blocks of quantized DCT coefficient values; and

quantizing each DCT coefficient value of each 2x2 sub-block of DCT coefficient values so as to provide corresponding 2x2 sub-blocks of quantized DCT coefficient values.

26. The method of Claim 25 further comprising the step of run-length encoding said ordered DCT coefficient values.

27. The method of Claim 24 wherein said coding format is a Huffman coding format.

28. A system for decoding a compressed image signal comprising:

separator means for receiving a compressed image signal representative of a selection value and coded values, separating said selection value and said coded values from one another, and providing an output of said separated selection value and said coded values;

decoder means for receiving and decoding according to a predetermined coding format said coded values, and for providing corresponding decoded values;

ordering means for receiving said decoded values and said selection value, ordering said decoded values according to a predetermined ordering format in response to said selection value, and providing an output forming a composite block of one of a block of discrete cosine transform coefficient values and various sub-blocks of discrete cosine transform coefficient values;

inverse discrete cosine transform means for receiving said composite block of discrete cosine transform coefficient values, performing an inverse discrete cosine transform operation on said one of said block and each sub-block of discrete cosine transform coefficient values of said composite block discrete cosine transform coefficient values, and providing an output of a corresponding one of a block of pixel data and sub-blocks of pixel data; and

combiner means for receiving said one of said block of pixel data and said sub-blocks of pixel data, receiving said selection value, and in accordance with said selection value combining when present said sub-blocks of pixel data into a combined block of pixel data and providing one of said block of pixel data and said combined block of pixel data as an output block of pixel data.

29. The system of Claim 28 wherein said decoder means comprises a Huffman decoder having an input coupled to said separator means and an output coupled to said ordering means.

30. The system of Claim 28 wherein said separator means provides said coded values as encoded non-zero coefficient values and run-length coefficient values and wherein said decoder means comprises:

a Huffman decoder comprising:

(a) a first decoder memory, storing Huffman decode values, having an input for receiving said non-zero coefficient values and an output; and

(b) a second decoder memory, storing Huffman code values, having an input for receiving said run-length coefficient values and an output; and

a run-length decoder having a non-zero coefficient value input coupled to said first decoder memory output, a run-length coefficient value input coupled to said decoder memory output and an output.

IPEA/US 12 MAY 1992
PCT/US 91/01350

31. The system of Claim 28 wherein said ordering means comprises an inverse zig-zag scan serializer having a coefficient input coupled to said decoder means, a control input for receiving said selection value and an output.

32. The system of Claim 30 wherein said ordering means comprises:

a coefficient buffer having an input coupled to said run-length decoder output and an output; and

an inverse zig-zag scan serializer having a coefficient input coupled to said coefficient buffer output, a control input for receiving said selection value and an output.

33. The system of Claim 28 wherein said ordering means provides an output of ones of a 16x16 block and 8x8, 4x4 and 2x2 sub-blocks of discrete cosine transform (DCT) coefficient values and wherein said inverse transform means comprises:

first inverse discrete cosine transform (IDCT) means for receiving and performing a 2x2 inverse discrete cosine transform (IDCT) operation on each received 2x2 sub-block of DCT coefficient values, and providing corresponding sub-blocks of 2x2 pixel data;

second IDCT transform means for receiving and performing a 4x4 IDCT operation on each received 4x4 sub-block of DCT coefficient values, and providing corresponding sub-blocks of 4x4 pixel data;

third IDCT transform means for receiving and performing an 8x8 IDCT operation on each received 8x8 sub-block of DCT coefficient values, and providing corresponding sub-blocks of 8x8 pixel data; and

fourth IDCT transform means for receiving and performing a 16x16 IDCT transform operation on said 16x16 block of DCT coefficient values, and providing a corresponding 16x16 block of pixel data.

34. The system of Claim 28 wherein said transform coefficient values are of quantized coefficient values, said system further comprising inverse quantization means disposed between said ordering means and said transform means for receiving and inverse quantizing transform

SUBSTITUTE SHEET
IPEA/US

coefficient values of said composite composite block of transform coefficient values, and for providing corresponding quantized transform coefficient values to said inverse transform means.

35. The system of Claim 33 wherein said 16x16 block and 8x8, 4x4 and 2x2 sub-blocks of discrete cosine transform (DCT) coefficient values are of quantized coefficient values, said system further comprising:

first memory means for, storing first inverse quantization values, receiving each quantized DCT coefficient value of each 2x2 sub-block of DCT coefficient values, and providing corresponding inverse quantized DCT coefficient values;

second memory means for, storing second inverse quantization values, receiving each quantized DCT coefficient value of each 4x4 sub-block of DCT coefficient values, and providing corresponding inverse quantized DCT coefficient values; and

third memory means for, storing third inverse quantization values, receiving each quantized DCT coefficient value of each 8x8 sub-block of DCT coefficient values, and providing corresponding inverse quantized DCT coefficient values; and

fourth memory means for, storing fourth inverse quantization values, for receiving each quantized DCT coefficient value of said 16x16 block of DCT coefficients value and providing corresponding inverse quantized DCT coefficient values.

36. A system for reconstructing a 16x16 block of pixel image data from a compressed image data signal comprising:

data separator means for, receiving a compressed image data signal containing in a predetermined order a selection value, coded run-length values, and coded non-zero discrete cosine transform (DCT) coefficient values indicative of a compressed 16x16 block of pixel data, separating said selection value, said coded run-length values, and said coded non-zero DCT coefficient values, and providing corresponding outputs of said selection value, said coded run-length values and said coded non-zero DCT coefficient values;

IPEA/US 12 MAY 199
PCT/US 91/0135

encoder means for, receiving and decoding according to a predetermined coding format each of said coded run-length values and said coded non-zero DCT coefficient values so as to provide decoded run-length values and non-zero DCT coefficient values, run-length decoding said run-length values and non-zero DCT coefficient values, and for providing an output of DCT coefficient values;

inverse serializer means for, receiving said DCT coefficient values and said selection value, ordering said DCT coefficient values according to a predetermined ordering format in response to said selection value so as to provide a composite block of ones of a 16x16 block of DCT coefficient values and 8x8, 4x4 and 2x2 sub-blocks of DCT coefficient values;

inverse quantization means for receiving and inverse quantizing each DCT coefficient of said composite block of DCT coefficient values, and for providing a corresponding output of a composite block of inverse quantized DCT coefficient values;

inverse transform means for, performing a 16x16 inverse discrete cosine transform (IDCT) operation on said composite block of inverse quantized DCT coefficient values when comprised of a 16x16 block of inverse quantized DCT coefficient values, performing an 8x8 IDCT operation on each 8x8 sub-block of inverse quantized DCT coefficient values when comprised of at least one 8x8 sub-block of inverse quantized DCT coefficient values, performing a 4x4 IDCT operation on each 4x4 sub-block of inverse quantized DCT coefficient values when comprised of at least one 4x4 block of inverse quantized DCT coefficient values, performing a 2x2 IDCT operation on each 2x2 sub-block of inverse quantized DCT coefficient values when comprised of at least one 2x2 block of inverse quantized DCT coefficient values, and for providing a corresponding ones of a 16x16 block and 8x8, 4x4 and 2x2 sub-blocks of pixel data;

sub-block combiner means for receiving said selection value, receiving when output from said inverse transform means each of said 16x16 block and 8x8, 4x4 and 2x2 sub-blocks of pixel data, assembling said 16x16 block and said 8x8, 4x4 and 2x2 sub-blocks of pixel data into an output 16x16 block of pixel data.

37. The system of Claim 36 wherein said inverse transform means comprises:

a 2x2 inverse discrete cosine transform (IDCT) element having an input for receiving 2x2 sub-blocks of quantized DCT coefficient values, and an output coupled to said sub-block combiner means;

a 4x4 IDCT element having an input for receiving 4x4 sub-blocks of quantized DCT coefficient values, and an output coupled to said sub-block combiner means;

an 8x8 IDCT element having an input for receiving 8x8 sub-blocks of quantized DCT coefficient values, and an output coupled to said sub-block combiner means; and

a 16x16 IDCT element having an input for receiving a 16x16 block of quantized DCT coefficient values, and an output coupled to said sub-block combiner means.

38. The system of Claim 37 wherein said inverse quantization means comprises:

first inverse quantizer means for receiving and inverse quantizing each 2x2 sub-block of quantized DCT coefficient values according to a first predetermined inverse quantization scheme, and for providing a corresponding 2x2 sub-block of inverse quantized DCT coefficient values to said 2x2 IDCT element input;

second inverse quantizer means for receiving and inverse quantizing each 4x4 sub-block of quantized DCT coefficient values according to a second predetermined inverse quantization scheme, and for providing a corresponding 4x4 sub-block of inverse quantized DCT coefficient values to said 4x4 IDCT element input;

third inverse quantizer means for receiving and inverse quantizing each 8x8 sub-block of quantized DCT coefficient values according to a third predetermined inverse quantization scheme, and for providing a corresponding 8x8 sub-block of inverse quantized DCT coefficient values to said 8x8 IDCT element input; and

fourth inverse quantizer means for receiving and inverse quantizing said 16x16 block of quantized DCT coefficient values according

to a fourth predetermined inverse quantization scheme, and for providing a corresponding 16x16 block of inverse quantized DCT coefficient values to said 16x16 IDCT element input.

39. The system of Claim 38 wherein said encoder means comprises:

a Huffman decoder comprising:

(a) a first decoder memory, storing Huffman decode values, having an input for receiving said coded non-zero coefficient values and an output; and

(b) a second decoder memory, storing Huffman code values, having an input for receiving said coded run-length coefficient values and an output; and

a run-length decoder having a non-zero coefficient value input coupled to said first decoder memory output, a run-length coefficient value input coupled to said decoder memory output and an output coupled to said inverse serializer means.

40. The system of Claim 39 further comprising a coefficient buffer disposed between said run-length decoder and said inverse serializer means, said coefficient buffer having an input coupled to said run-length decoder output and an output coupled to said inverse serializer means.

41. A method for reconstructing a block of pixel data from a compressed image data signal comprising the steps of:

receiving a compressed image data signal representative of a selection value and coded values;

separating said selection value and said coded values from one another so as to provide an output of said separated selection value and said coded values;

decoding according to a predetermined coding format said coded values so as to provide corresponding decoded values;

ordering said decoded values according to a

predetermined ordering format according to said selection value so as to provide an output forming a composite block of one of a block of discrete cosine transform coefficient values and various sub-blocks of discrete cosine transform coefficient values;

performing an inverse discrete cosine transform operation on said one of said block and each sub-block of discrete cosine transform coefficient values of said composite block of discrete cosine transform coefficient values to provide an output of a corresponding one of a block of pixel data and sub-blocks of pixel data; and

combining in accordance with said selection value said sub-blocks of pixel data, when present, into a combined block of pixel data so as to provide one of said block of pixel data and said combined block of pixel data as an output block of pixel data.

42. The method of Claim 41 wherein said step of decoding comprises the steps of:

storing Huffman decode values in a memory;

addressing said memory with said coded values; and

providing a decoded value output from said memory corresponding to each input coded value.

43. The method of Claim 41 wherein said step of ordering said decoded values comprises the step inverse zig-zag scan serializing said decoded values.

44. The method of Claim 41 wherein said step of ordering said decoded values provides said composite block of transform coefficient values as ones of a 16x16 block and 8x8, 4x4 and 2x2 sub-blocks of discrete cosine transform (DCT) coefficient values and wherein said step of performing an inverse transform operation on said one of said block and each sub-block of transform coefficient values comprises the steps of:

performing a 2x2 inverse discrete cosine transform (IDCT) operation on each received 2x2 sub-block of DCT coefficient values so as to provide corresponding sub-blocks of 2x2 pixel data;

performing a 4x4 IDCT operation on each received 4x4 sub-block of DCT coefficient values so as to provide corresponding sub-blocks of 4x4 pixel data;

IPEA/US 12 MAY 1992
PCT/US 91/01350

performing an 8x8 IDCT operation on each received 8x8 sub-block of DCT coefficient values so as to provide corresponding sub-blocks of 8x8 pixel data; and

performing a 16x16 IDCT transform operation on said 16x16 block of DCT coefficient values so as to provide a corresponding 16x16 block of pixel data.

45. The method of Claim 41 further comprising the step of inverse quantizing each transform coefficient value of said composite block of transform coefficient values.

46. The method of Claim 43 further comprising the step of inverse quantizing each transform coefficient value of said composite block of transform coefficient values.

47. A method for reconstructing a 16x16 block of pixel image data from a compressed image data signal comprising:

receiving a compressed image data signal containing in a predetermined order a selection value, coded run-length values, and coded non-zero discrete cosine transform (DCT) coefficient values indicative of a compressed 16x16 block of pixel data;

separating said selection value, said coded run-length values, and said coded non-zero DCT coefficient values so as to provide corresponding outputs of said selection value, said coded run-length values and said coded non-zero DCT coefficient values;

decoding according to a predetermined coding format each of said coded run-length values and said coded non-zero DCT coefficient values so as to provide decoded run-length values and non-zero DCT coefficient values;

run-length decoding said run-length values and non-zero DCT coefficient values so as to provide an output of DCT coefficient values;

ordering said DCT coefficient values according to a predetermined ordering format in response to said selection value so as provide a composite block of ones of a 16x16 block of DCT coefficient values and 8x8, 4x4 and 2x2 sub-blocks of DCT coefficient values;

SUBSTITUTE SHEET
IPEA/IS

inverse quantizing each DCT coefficient of said composite block of DCT coefficient values so as to provide a corresponding output of a composite block of inverse quantized DCT coefficient values;

performing a 16x16 inverse discrete cosine transform (IDCT) operation on said composite block of inverse quantized DCT coefficient values when comprised of a 16x16 block of inverse quantized DCT coefficient values so as to provide a corresponding 16x16 block of pixel data;

performing an 8x8 IDCT operation on each 8x8 sub-block of inverse quantized DCT coefficient values when comprised of at least one 8x8 block of inverse quantized DCT coefficient values so as to provide corresponding 8x8 sub-blocks of pixel data;

performing a 4x4 IDCT operation on each 4x4 sub-block of inverse quantized DCT coefficient values when comprised of at least one 4x4 block of inverse quantized DCT coefficient values so as to provide corresponding 4x4 sub-blocks of pixel data;

performing a 2x2 IDCT operation on each 2x2 sub-block of inverse quantized DCT coefficient values when comprised of at least one 2x2 block of inverse quantized DCT coefficient values so as to provide corresponding 2x2 sub-blocks of pixel data;

combining in response to selection value said 8x8, 4x4 and 2x2 sub-blocks of pixel data into a composite 16x16 block of pixel data so as to provide one of said 16x16 block of pixel data and said composite 16x16 block of pixel data as a reconstructed 16x16 block of pixel data.

48. The method of Claim 47 wherein said step of inverse quantizing comprises the steps of:

inverse quantizing each 2x2 sub-block of quantized DCT coefficient values according to a first predetermined quantization scheme so as to provide a corresponding 2x2 sub-block of quantized DCT coefficient values;

inverse quantizing each 4x4 sub-block of quantized DCT coefficient values according to a second predetermined inverse quantization scheme so as to provide a corresponding 4x4 sub-block of inverse quantized DCT coefficient values;

SUBSTITUTE SHEET
IPEA/US

inverse quantizing each 8x8 sub-block of quantized DCT coefficient values according to a third predetermined inverse quantization scheme so as to provide a corresponding 8x8 sub-block of inverse quantized DCT coefficient values; and

inverse quantizing said 16x16 block of quantized DCT coefficient values according to a fourth predetermined inverse quantization scheme so as to provide a corresponding 16x16 block of inverse quantized DCT coefficient values.

49. The method of Claim 47 wherein said step of decoding according to a predetermined coding format each of said coded run-length values and said coded non-zero DCT coefficient values comprises the steps of:

Huffman decoding said coded run-length values so as to provide said decoded run-length values; and

Huffman decoding said said coded non-zero DCT coefficient values so as to provide said non-zero DCT coefficient values.

50. The method of Claim 48 wherein said step of decoding according to a predetermined coding format each of said coded run-length values and said coded non-zero DCT coefficient values comprises the steps of:

Huffman decoding said coded run-length values so as to provide said decoded run-length values; and

Huffman decoding said said coded non-zero DCT coefficient values so as to provide said non-zero DCT coefficient values.

51. The method of Claim 47 wherein said step ordering said DCT coefficient values according to a predetermined ordering format comprises the step of inverse zig-zag scan serializing said DCT coefficients wherein said selection value is indicative of DCT coefficients forming each 16x16 block of DCT coefficient values and each 8x8, 4x4 and 2x2 sub-blocks of DCT coefficient values of said composite block of DCT coefficient values.

IPEA/US 12 MAY 1992
PCT/US 91/01350

52. The method of Claim 50 wherein said step ordering said DCT coefficient values according to a predetermined ordering format comprises the step of inverse zig-zag scan serializing said DCT coefficients wherein said selection value is indicative of DCT coefficients forming each 16x16 block of DCT coefficient values and each 8x8, 4x4 and 2x2 sub-blocks of DCT coefficient values of said composite block of DCT coefficient values.

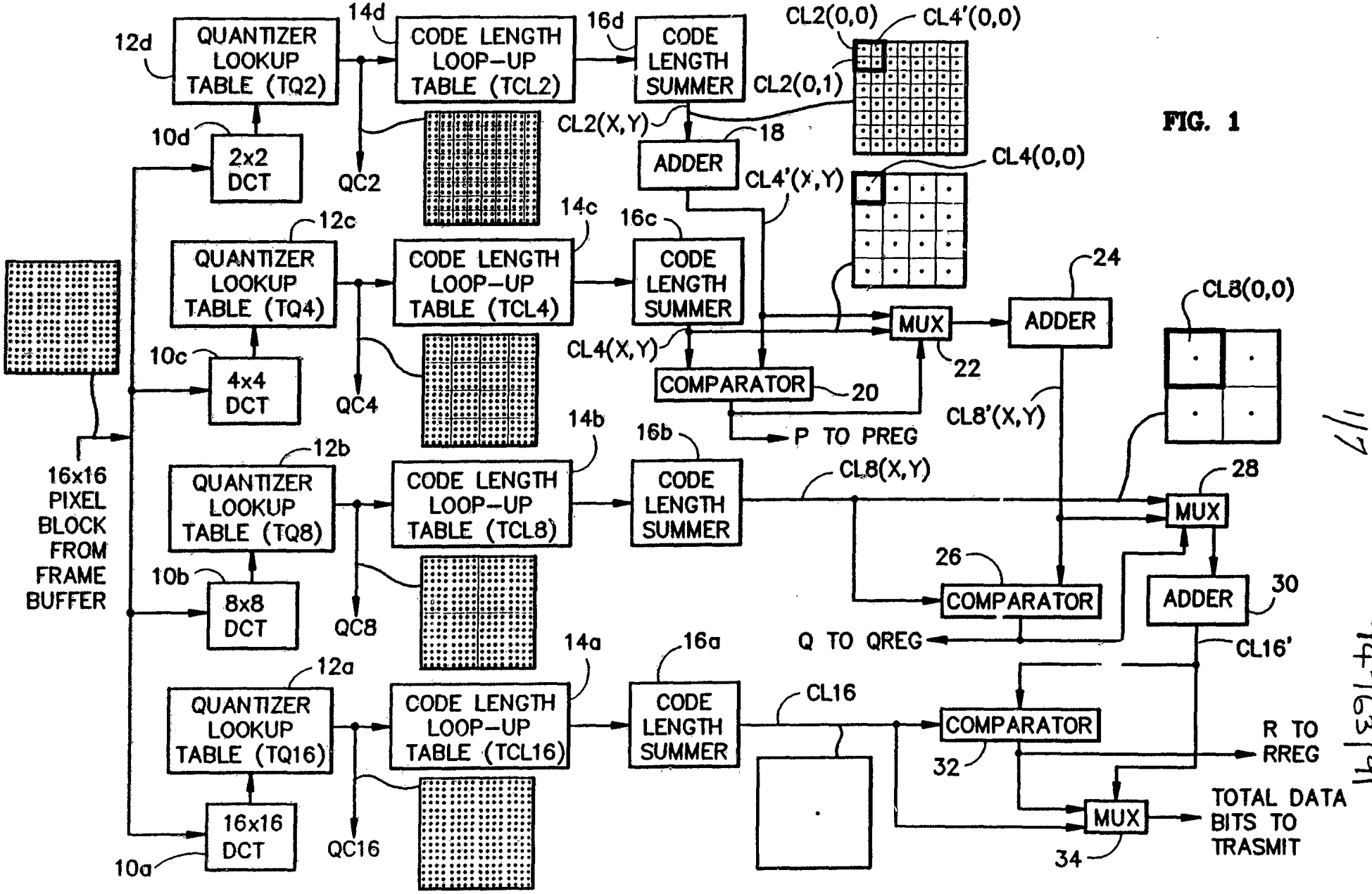

12d
QUANTIZER LOOKUP TABLE (TQ2)
14d
CODE LENGTH LOOP-UP TABLE (TCL2)
16d
CODE LENGTH SUMMER
CL2(0,0)
CL4'(0,0)
CL2(0,1)
CL2(X,Y)
18
ADDER
CL4'(X,Y)
CL4(0,0)
10d
2x2 DCT
QC2
12c
QUANTIZER LOOKUP TABLE (TQ4)
14c
CODE LENGTH LOOP-UP TABLE (TCL4)
16c
CODE LENGTH SUMMER
24
MUX
ADDER
CL8(0,0)
CL4(X,Y)
22
10c
4x4 DCT
QC4
COMPARATOR
20
CL8'(X,Y)
P TO PREG
14b
16b
12b
16x16 PIXEL BLOCK FROM FRAME BUFFER
QUANTIZER LOOKUP TABLE (TQ8)
CODE LENGTH LOOP-UP TABLE (TCL8)
CODE LENGTH SUMMER
CL8(X,Y)
28
MUX
26
10b
8x8 DCT
QC8
COMPARATOR
ADDER
30
Q TO QREG
CL16'
14a
16a
12a
QUANTIZER LOOKUP TABLE (TQ16)
CODE LENGTH LOOP-UP TABLE (TCL16)
CODE LENGTH SUMMER
CL16
COMPARATOR
R TO RREG
32
MUX
TOTAL DATA BITS TO TRASMIT
34
16x16 DCT
QC16
10a

FIG. 1

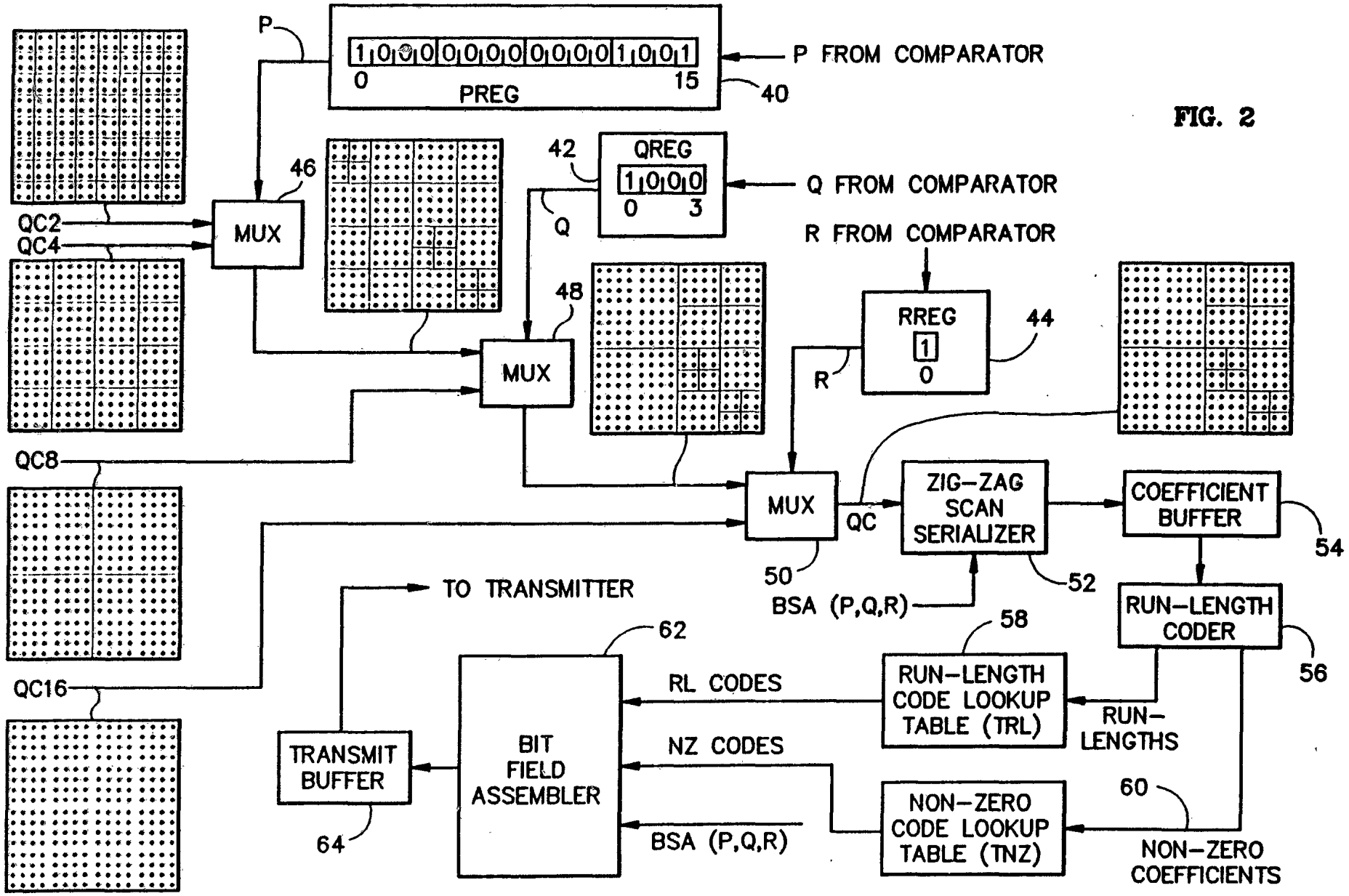

P
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0
PREG
15
40
P FROM COMPARATOR
42
QREG
1 0 0 0
0 3
Q FROM COMPARATOR
Q
R FROM COMPARATOR
46
MUX
QC2
QC4
48
MUX
RREG
1
0
44
R
QC8
MUX
QC
ZIG–ZAG SCAN SERIALIZER
COEFFICIENT BUFFER
54
50
BSA (P,Q,R)
52
TO TRANSMITTER
62
58
RUN–LENGTH CODER
56
QC16
RL CODES
RUN–LENGTH CODE LOOKUP TABLE (TRL)
RUN–LENGTHS
TRANSMIT BUFFER
BIT FIELD ASSEMBLER
NZ CODES
60
NON–ZERO CODE LOOKUP TABLE (TNZ)
64
BSA (P,Q,R)
NON–ZERO COEFFICIENTS

FIG. 2

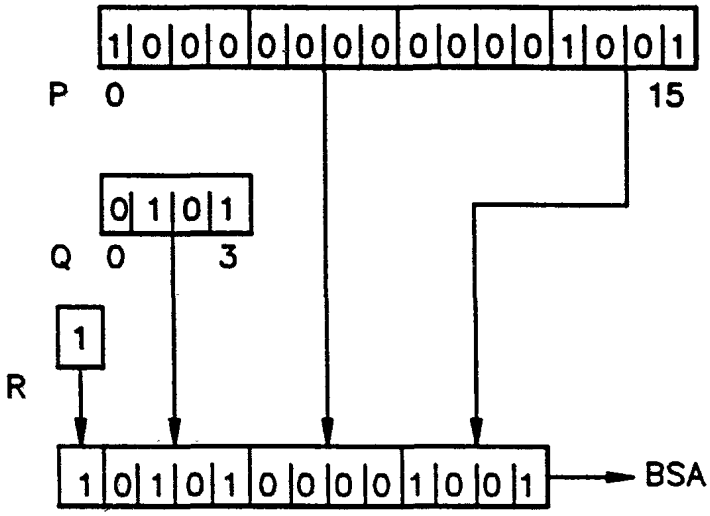


FIG. 3a

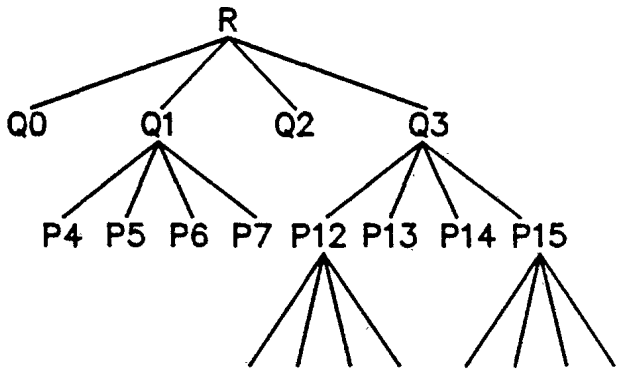


FIG. 3b

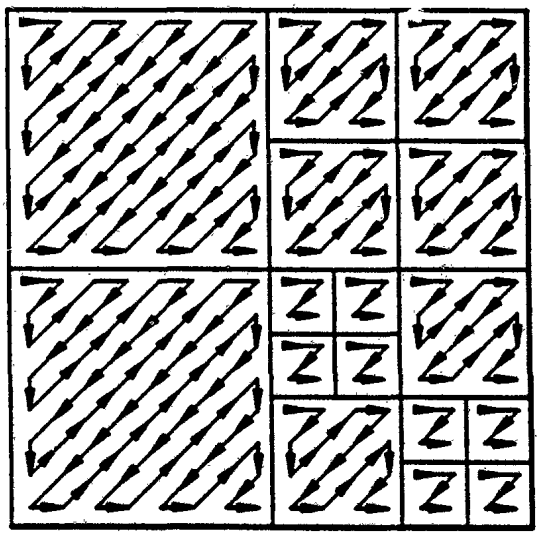


FIG. 4a

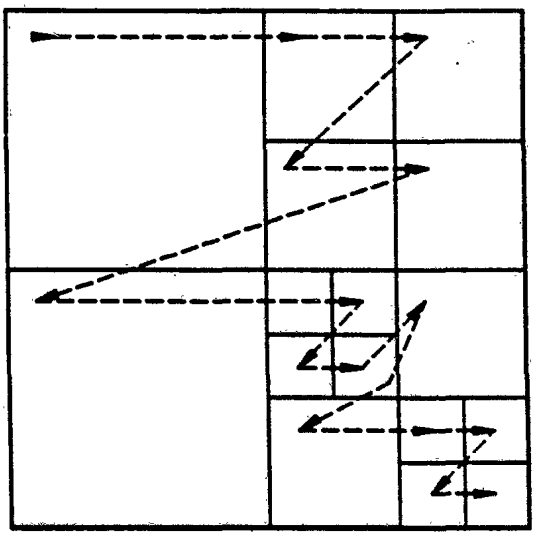


FIG. 4b

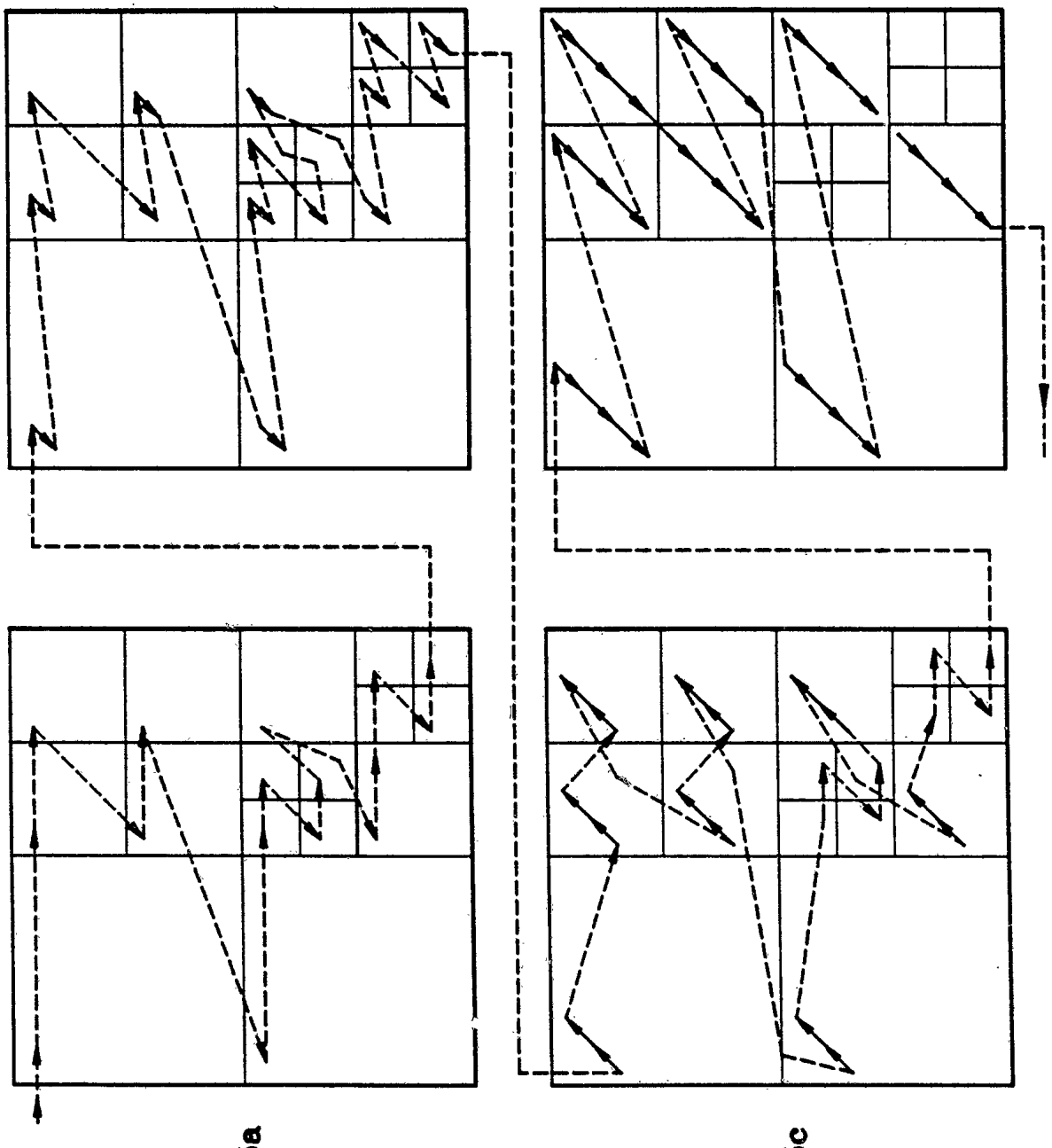


FIG. 5b

FIG. 5d

FIG. 5a

FIG. 5c

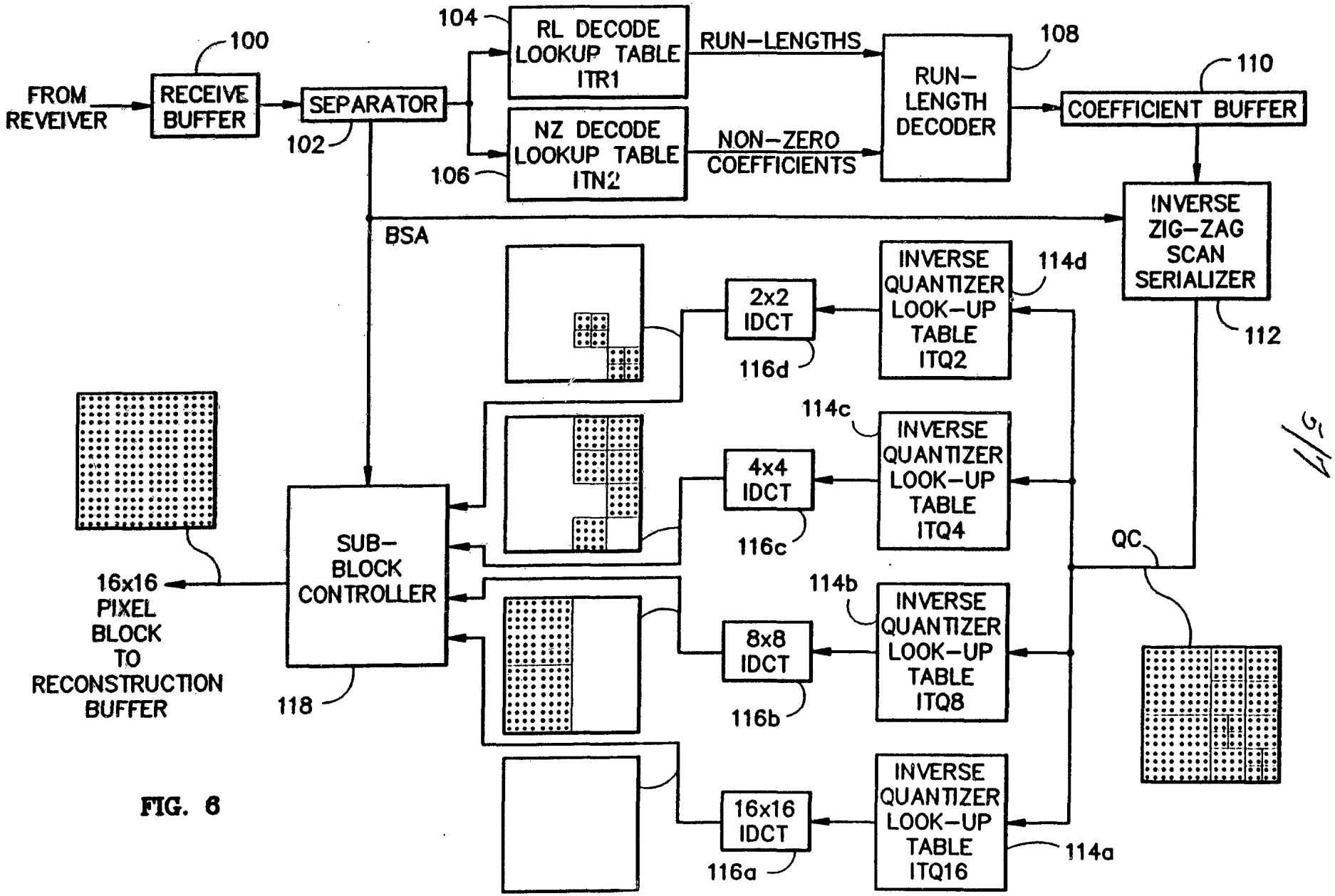

FROM REVEIVER
RECEIVE BUFFER
100
SEPARATOR
102
104
RL DECODE LOOKUP TABLE ITR1
106
NZ DECODE LOOKUP TABLE ITN2
RUN-LENGTHS
NON-ZERO COEFFICIENTS
RUN-LENGTH DECODER
108
COEFFICIENT BUFFER
110
INVERSE ZIG-ZAG SCAN SERIALIZER
112
BSA
INVERSE QUANTIZER LOOK-UP TABLE ITQ2
114d
2x2 IDCT
116d
INVERSE QUANTIZER LOOK-UP TABLE ITQ4
114c
4x4 IDCT
116c
INVERSE QUANTIZER LOOK-UP TABLE ITQ8
114b
8x8 IDCT
116b
INVERSE QUANTIZER LOOK-UP TABLE ITQ16
114a
16x16 IDCT
116a
QC
SUB-BLOCK CONTROLLER
118
16x16 PIXEL BLOCK TO RECONSTRUCTION BUFFER

FIG. 6

5/14

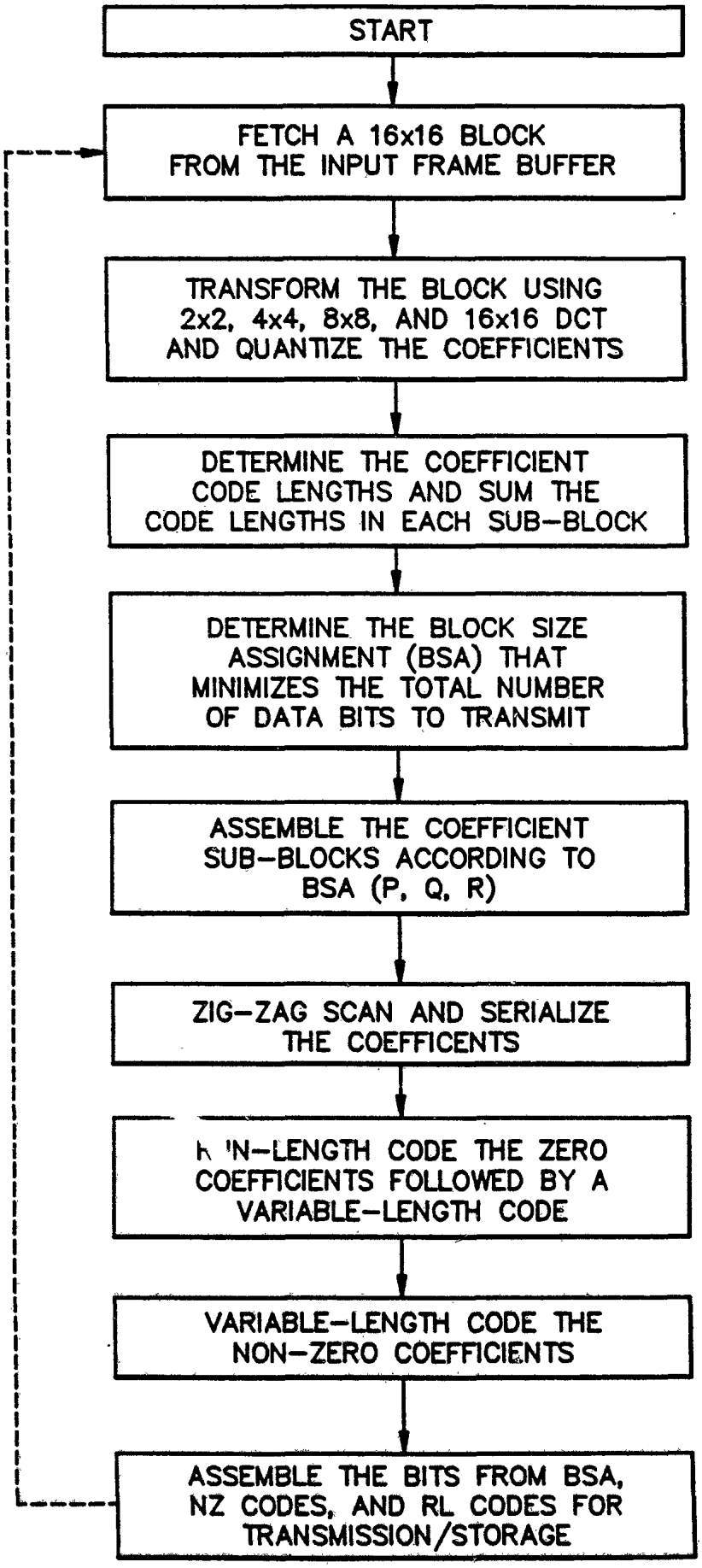

START
FETCH A 16x16 BLOCK FROM THE INPUT FRAME BUFFER
TRANSFORM THE BLOCK USING 2x2, 4x4, 8x8, AND 16x16 DCT AND QUANTIZE THE COEFFICIENTS
DETERMINE THE COEFFICIENT CODE LENGTHS AND SUM THE CODE LENGTHS IN EACH SUB-BLOCK
DETERMINE THE BLOCK SIZE ASSIGNMENT (BSA) THAT MINIMIZES THE TOTAL NUMBER OF DATA BITS TO TRANSMIT
ASSEMBLE THE COEFFICIENT SUB-BLOCKS ACCORDING TO BSA (P, Q, R)
ZIG-ZAG SCAN AND SERIALIZE THE COEFFICENTS
RUN-LENGTH CODE THE ZERO COEFFICIENTS FOLLOWED BY A VARIABLE-LENGTH CODE
VARIABLE-LENGTH CODE THE NON-ZERO COEFFICIENTS
ASSEMBLE THE BITS FROM BSA, NZ CODES, AND RL CODES FOR TRANSMISSION/STORAGE

FIG. 7

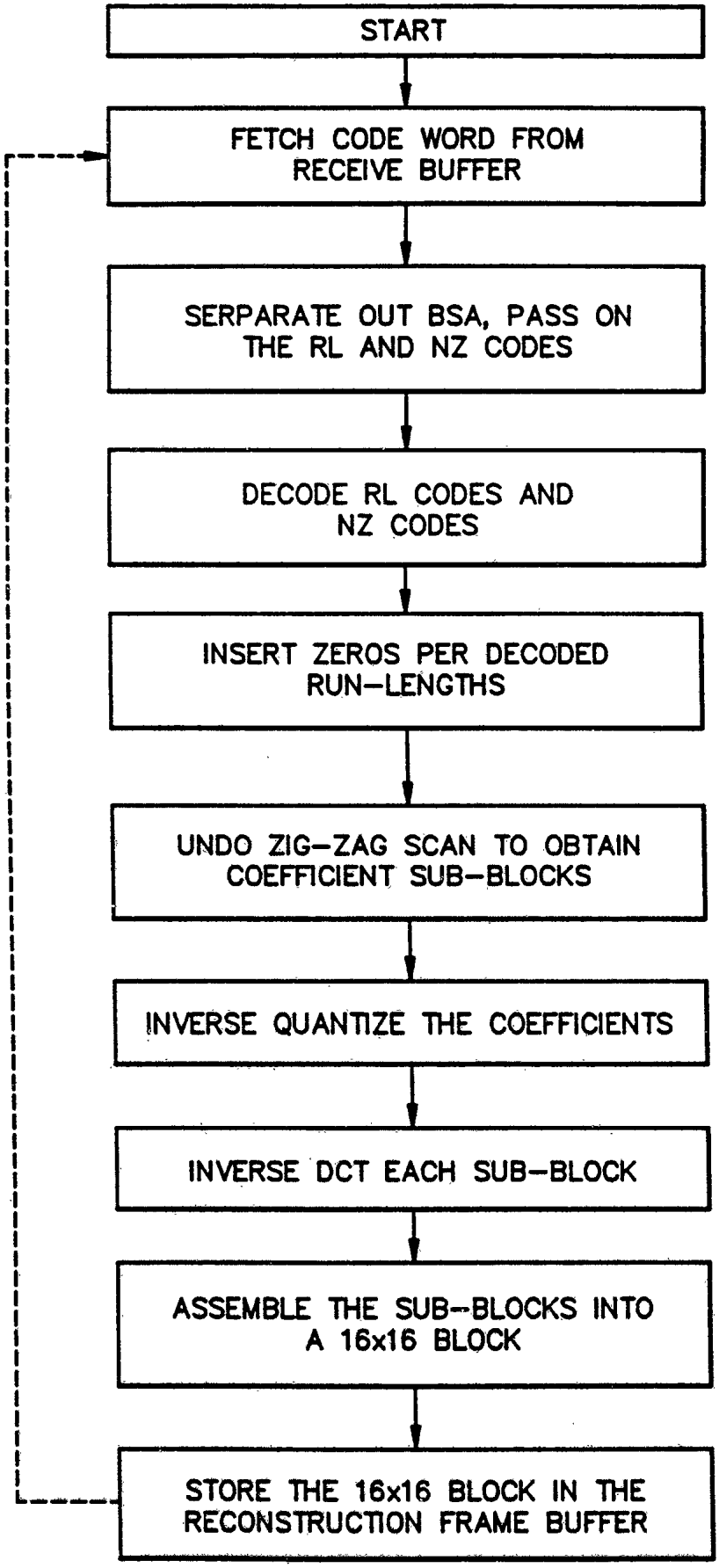

START
FETCH CODE WORD FROM RECEIVE BUFFER
SERPARATE OUT BSA, PASS ON THE RL AND NZ CODES
DECODE RL CODES AND NZ CODES
INSERT ZEROS PER DECODED RUN-LENGTHS
UNDO ZIG-ZAG SCAN TO OBTAIN COEFFICIENT SUB-BLOCKS
INVERSE QUANTIZE THE COEFFICIENTS
INVERSE DCT EACH SUB-BLOCK
ASSEMBLE THE SUB-BLOCKS INTO A 16x16 BLOCK
STORE THE 16x16 BLOCK IN THE RECONSTRUCTION FRAME BUFFER

FIG. 8

INTERNATIONAL SEARCH REPORT

International Application No PCT/US91/01350

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC(5) H04N 1/415, G06K 9/36
US CL 358/432, 433, 382/41

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
US	358/432, 433, 133, 138, 382/41, 42

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [6]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
A	US,A 4,504,860 (NICOL et al) 12 March 1985	1-5
A	US,A 4,179,709 (WORKMAN) 18 December 1979	1-5
A	US,A 4,293,920 (MEROLA) 06 October 1981	1-5
A	US,A 4,807,042(TAMAKA) 21 February 1989	1-5
A	US,A 4,774,574 (DALY et al) 27 September 1988	1-5
A,P	US,A 4,922,341 (STROBACH) 01 May 1990	1-5
A,P	US,A 4,924,309 (HARTNACK et al) 08 May 1990	1-5

* Special categories of cited documents: [15]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search [2]	Date of Mailing of this International Search Report [2]
20 May 1991	25 JUN 1991
International Searching Authority [1]	**Signature of Authorized Officer**
ISA/US	NGUYEN NGOC-HO INTERNATIONAL DIVISION for Edward L. Coles, Sr. Ngoc-Ho Nguyen

Form PCT/ISA/210 (second sheet) (May 1986)

FURTHER INFORMATION CONTINUED FROM THE SECOND SHEET

A	Randall, C. Reininger ET AL, "DISTRIBUTIONS OF TWO-DIMENSIONAL DCT Coefficients for Images", IEEE Transactions on Communications, Vol. Com-31, No.6, pages 835-839 June 1983.	
A	WEN-HSIUNG GHEN ET AL, "Scene Adaptive Coder" IEEE Transactions on Communications, VOL-COM 32, No.3, March 1984, pages 225-232.	
A	David R. Comstock et al, "Hamming Coding of DCT-Compressed Images over Noisy Channels", IEEE Transactions on Communications, Vol. Comm-32 No.7 July 1984 pages 856-861.	

V. ☐ OBSERVATIONS WHERE CERTAIN CLAIMS WERE FOUND UNSEARCHABLE [1]

This international search report has not been established in respect of certain claims under Article 17(2) (a) for the following reasons:

1. ☐ Claim numbers . , because they relate to subject matter [1] not required to be searched by this Authority, namely:

2. ☐ Claim numbers..... , because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out [1], specifically:

3. ☐ Claim numbers_____, because they are dependent claims not drafted in accordance with the second and third sentences of PCT Rule 6.4(a).

VI. ☐ OBSERVATIONS WHERE UNITY OF INVENTION IS LACKING [2]

This International Searching Authority found multiple inventions in this international application as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims of the international application.

2. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims of the international application for which fees were paid, specifically claims:

3. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim numbers:

4. ☐ As all searchable claims could be searched without effort justifying an additional fee, the International Searching Authority did not invite payment of any additional fee.

Remark on Protest

☐ The additional search fees were accompanied by applicant's protest.

☐ No protest accompanied the payment of additional search fees.

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No [18]
A	N. Ahmed et al, "Discrete Cosine Transform", IEEE Transactions on Computers, January 1974, pages 90-93	
A	Wen-Hsiung Chen et al "Adaptive Coding of Monochrome and Color Images, IEEE Transactions on communications, Col COM-25 No.11 November 1977 pages 1285-1292	
A	John A. Roese et al "interframe Cosine Transform Image Coding" IEEE Transactions on Communications Vol COM-25 No.11 November 1977 pages 1329-1339	