(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2001 (30.08.2001)

PCT

(10) International Publication Number
WO 01/63772 A1

(51) International Patent Classification[7]: H03M 7/34, H04N 1/415

(21) International Application Number: PCT/US01/05722

(22) International Filing Date: 22 February 2001 (22.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/513,309 25 February 2000 (25.02.2000) US

(71) Applicant: PHYSICAL OPTICS CORPORATION [US/US]; 20600 Gramercy Place, Building 100, Torrance, CA 90501-1821 (US).

(72) Inventors: TERNOVSKIY, Igor, V.; 5700 Ravensbur Drive, #209, Rancho Palos Verdes, CA 90275 (US). DEVIVYE, Aleksandr, A.; 6517 Kester Avenue, #4, Van Nuys, CA 91411 (US). ROTENBERG, Joseph; 719 Price Drive, Burbank, CA 91504 (US). LIN, Freddie; 800 Calle De Arboles, Redondo Beach, CA 90277 (US).

(74) Agents: NILLES, Andrew, J. et al.; Nilles & Nilles, S.C., Firstar Center, Suite 2000, 777 East Wisconsin Avenue, Milwaukee, WI 53202 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR OPTIMIZED LOSSLESS COMPRESSION USING A PLURALITY OF CODERS

30

32

	ARITHMETIC	LEMPEL-ZIV	HUFFMAN	...	n
8	LOSSLESS CODER (Ar,8)	LOSSLESS CODER (LZ,8)			LOSSLESS CODER (n,8)
10	LOSSLESS CODER (Ar,10)				
...					
m	LOSSLESS CODER (Ar,M)				LOSSLESS CODER (n,m)

(57) Abstract: A method of lossless compression of a stream of data first includes using a plurality of lossless coders to compress a test portion of the data stream (30). Once the test portion is compressed, the method determines a performance characteristic(s) associated with each of the lossless coders (32). Then the method selects one of the lossless coders based on the performance characteristic(s) and encodes a first portion of the data stream with the selected coder. Thereafter, the method includes repeating the using, determining, selecting and encoding steps for another test portion and a second portion of the data stream. Notably, the repeating step may include selecting a different one of the lossless coders.

WO 01/63772 A1

METHOD AND APPARATUS FOR OPTIMIZED LOSSLESS COMPRESSION USING A PLURALITY OF CODERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention is directed to data compression techniques and, more particularly, to a method and apparatus for selecting among different types of lossless compression coders to optimize system performance.

2. Description of the Related Art

Data compression operates to minimize the number of bits used to store or transmit information and encompasses a wide array of software and hardware compression techniques. Notably, depending on the type of data to be compressed and any number of other factors, particular compression techniques can provide markedly superior performance in terms of compression ratio and coding speed.

Generally, data compression includes taking a stream of symbols or phrases and converting them into codes that are smaller (in bit length) than the original data. Known compression techniques and algorithms can be divided into two major families including lossy and lossless. Lossy data compression can be used to greatly increase data compression ratios; however, increased compression comes at the expense of a certain loss in accuracy. As a result, lossy compression typically is implemented only in those instances in which some data loss is acceptable. For example, lossy compression is used effectively used when applied to digitized voice signals and graphics images. Lossless compression, on the other hand, is a family of data compression that utilizes techniques designed to generate an exact duplicate of the input data stream after a compression/decompression cycle. This type of compression is necessary when storing database records, word processing files, etc., where loss of information is absolutely unacceptable. The present invention is directed to lossless data compression.

Some lossless compression algorithms use information theory to generate variable length codes when given a probability table for a given set of symbols. The decision to output a certain code for a particular symbol or set of symbols (i.e., message) is based on a model. The model is a set of rules used to process input messages, and in response, determine which codes to output. An algorithm or program uses the model to analyze the symbols (e.g., determine a probability associated with the symbol) and then outputs an appropriate code based on that processing. There are any number of ways to model data, all of which can use the same coding technique to produce their output. In general, to compress data efficiently, a model should be selected that predicts symbols or phrases with high probabilities because symbols or messages that have a high probability have a low information content, and therefore require fewer bits to encode. The next step is to encode the symbols using a particular lossless coder.

Conventionally, lossless compression coders can be grouped according to whether they implement statistical modeling or dictionary-based modeling. Statistical modeling reads and encodes a single symbol at a time using the probability of the character's appearance, while dictionary-based modeling uses a single code to replace strings of symbols. Notably, in dictionary-based modeling, the model is significantly more important than in statistical-based modeling because problems associated with encoding every symbol are significantly reduced.

One form of statistical data compression is known as Shannon-Fano (S-F) coding. S-F coding was developed to provide variable-length bit coding so as to allow coding symbols with exactly (or a close approximation to) the number of bits of information that the message or symbol contains. S-F coding relies on knowing the probability of each symbol's appearance in a message. After the probabilities are determined, a table of codes is constructed with each code having a different number of bits (advantageously, symbols with low probabilities have more bits). One problem with a coding technique such as this is that it creates variable length codes that have an integral number of bits, even though the information to be coded likely will require a non-integral number of bits.

Another type of coding, Huffman coding, is similar to S-F coding in that it creates variable length codes that are an integral number of bits, but it utilizes a completely different algorithm. Generally, S-F and Huffman codings are close in performance but Huffman coding, it has been determined, always at least equals the efficiency of S-F coding so it is therefore preferred, especially since both algorithms take a similar amount of processing power. Although Huffman coding is relatively easy to implement and economical for both coding and decoding, it is inefficient due to its use of an integral number of bits per code as in S-F coding. If a particular symbol is determined to have an information content (i.e., entropy) of 1.5 bits, a Huffman coder will generate a code having a bit count that is either one or two bits. Generally, if a statistical method could assign a 90% probability to a given symbol, the optimal code size would be 0.15 bits; however, Huffman or S-F coding likely would assign a one bit code to the symbol, which is six times larger than necessary.

In view of this problem associated with utilizing an integral number of bits, arithmetic coding was developed. Arithmetic coding replaces a stream of input symbols with a single floating point output number, and bypasses the step of replacing an input symbol with a specific code. Because an arithmetic code is not limited to being optimal only when the symbol probabilities are integral powers of one-half (which is most often not the case), it attains the theoretical entropy of the symbol to be coded, thus maximizing compression efficiency for any known source. In other words, if the entropy of a given character is 1.5 bits, arithmetic coding uses 1.5 bits to encode the symbol, an impossibility for Huffman and Shannon-Fano coding. Although arithmetic coding is extremely efficient, it consumes rather large amounts of computing resources, both in terms of CPU power and memory. This is due to the fact that sophisticated models that demand a significant amount of memory must be built, and that the algorithm itself requires a significant amount of computational operations.

In an alternative to the above types of lossless coding, known as substitutional or dictionary-based coding, dictionary-based compression algorithms replace occurrences of particular phrases (i.e., groups of bytes) in a data stream with a reference to a previous occurrence of those phrases. Unlike the above systems that achieve compression by encoding symbols into bit strings that use fewer bits than the original symbols, dictionary-based algorithms do not encode single symbols. Rather, dictionary-based compression techniques encode variable length strings of symbols as single "tokens." It is these tokens that form an index to a phrase dictionary. Because the tokens are smaller than the phrases they replace, compression occurs. Two main classes of dictionary-based compression schemes are known as the LZ77 and LZ78 compression algorithms of the Lempel-Ziv family of compression coders. Notably, dictionary-based coding is utilized extensively in desktop general purpose compression and has been implemented by Compuserve Information Service to encode bit-mapped graphical images. For example, the GIF format uses a LZW variant to compress repeated sequences and screen images. Although dictionary-based compression techniques are very popular forms of compression, the disadvantage of such algorithms is that a more sophisticated data structure is needed to handle the dictionary.

Overall, as communication mediums such as the internet expand, data compression will continue to be extremely important to the efficient communication of data, with different compression algorithms providing particular advantages in particular arenas. There are many types of data compression methods that are being implemented in the art, including those described above as well as others. In addition, there are many variants associated with each type of known compression algorithm and many improvements have been developed. Again, depending on any number of factors associated with the system and the type of data being compressed, each may be preferred to provide optimum data encoding.

Because different ones of known coding techniques provide unique benefits depending upon various operational factors including the data to be encoded, a lossless compression system that selectively encodes data with different types of coders was desired. The telecommunications industry, in particular, is in need of a system which implements different types of coders, especially when the incoming data is received from multiple sources that provide different types of unknown data, i.e., when different portions of the data stream would be optimally compressed with different coding techniques.

SUMMARY OF THE INVENTION

The present invention is directed to a method and apparatus that determines which of a number of embedded coding schemes will optimally compress different portions of an incoming data stream. The method of the preferred embodiment is designed to accommodate a data stream characterized by having different packets of information (e.g., from sources unknown to the encoders) each of which may have different associated statistics.

According to a first aspect of the preferred embodiment, a method of lossless compression of a stream of data includes providing a plurality of lossless coders. The method then includes selecting one of the lossless coders to compress the stream of data, and thereafter encoding the data stream with the selected lossless coder.

According to another aspect of the preferred embodiment, a method of lossless compression of a stream of data includes using a plurality of lossless coders to compress a test portion of the data stream. Once the test portion is compressed, the method determines a performance characteristic associated with each of the lossless coders. Then the method includes selecting one of the lossless coders based on the determining step and encoding a first portion of the data stream with the selected coder. Thereafter, the method includes repeating the using, determining, selecting and encoding steps for another test portion and a second portion of the data stream. Notably, the repeating step may include selecting a different one of the lossless coders.

According to a further aspect of the preferred embodiment, each of the lossless coders uses (1) a compression technique, and (2) a number of bits per word determined by the selecting step, in the encoding step. And, the compression technique is one of Arithmetic coding, Huffman coding and LZ coding.

According to yet another aspect of the preferred embodiment, an apparatus for lossless data compression includes an interface to receive a stream of data. In addition, the apparatus includes a plurality of lossless coders and a processor. In operation, each lossless coder separately compresses a test portion of the data stream and, in response, the processor determines a performance characteristic associated with each of the lossless coders, and then selects, based on the performance characteristics, one of the lossless coders to encode at least a first portion of the data stream.

According to a still further aspect of the preferred embodiment, the performance characteristic includes at least one of compression ratio and duration of the compression of the test portion for a corresponding lossless coder. Moreover, the encoder includes a plurality of processors and each of the lossless coders corresponds to one of the processors, and wherein the lossless coders compress the same test portion in parallel.

These and other objects, advantages, and features of the invention will become apparent to those skilled in the art from the detailed description and the accompanying drawings. It should be understood, however, that the detailed description and accompanying drawings, while indicating preferred embodiments of the present invention, are given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the present invention without departing from the spirit thereof, and the invention includes all such modifications.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred exemplary embodiments of the invention are illustrated in the accompanying drawings in which like reference numerals represent like parts throughout, and in which:

FIG. 1 is a flow diagram showing the general operation of a method of the preferred embodiment;

FIG. 1A is a chart showing an array of lossless coders used in the method shown in FIG. 1;

FIG. 2 is a generic block diagram showing an encoding/decoding system of the preferred embodiment; and

FIG. 3 is a schematic diagram showing the data stream as it is encoded/decoded by the system shown in FIG. 2.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Fig. 1, a method 10 includes, after initialization and start-up at Step 12, inputting data to the system at Step 14. The data input at Step 14 may be synchronous or asynchronous data. Notably, the data stream may be received from unspecified sources such as sensors that monitor temperature, pressure, etc. of a subject (e.g., telemetric data gathered in military applications) and that continuously transmit readings to the system encoder (described below) of the preferred embodiment. Unspecified data necessarily implies that the statistics associated with the data are random, and therefore unlike known systems that perform compression with a single type of encoder based on knowledge regarding the statistics of the data, the preferred embodiment is able to efficiently code a data stream comprised of different types of data. Other types of applications where this type of random data may originate from multiple sources include hospital monitoring applications, chemical factories, nuclear plants, and others.

As the data is continuously input to the system, it is transmitted to a division on communication block where method 10, at Step 16, processes the data by dividing or framing the data for further communication thereof. The division on communication block is implemented by method 10 in conventional fashion. Next, at Step 18, the data is pre-processed which may include generating a histogram indicating the statistics associated with the data framed in Step 16.

Once the data has been pre-processed at Step 18, method 10 adds synchronization and header codes at Step 20, as required to further process and identify the data bits in the data stream. Upon completion of Step 20, the data is transmitted to a plurality of coders that provide lossless compression. In particular, at Step 22, method 10 codes a test portion of the data stream with a plurality of lossless coders and determines system performance criteria associated with each of the coders. The coders used to code the portion of the data in Step 22 are shown at 32 in chart 30 of Fig. 1A. The columns of the charts indicate different types of lossless coding techniques/algorithms which may include Huffman coding, Arithmetic coding, Lempel-Ziv coding, as well as variants of these and other known coding techniques. Notably, the method also compares the output of the coding techniques with the data stream without encoding/compressing because, in certain circumstances, uncompressed data may be optimum.

In general, the columns comprise lossless coding techniques. The rows comprise different designations for the number of bits per word, bpw 1-m, may be used to encode the data. For instance, the bits per word associated with the interface may be set to be eight bits, ten bits, etc., for example. As a result, at Step 22, method 10 codes a portion of the data with *n x m* number of lossless coders. Preferably, Step 22 is performed for a test period of time or amount of data to determine which of the lossless coders achieves optimum system performance prior. Thereafter, the data is encoded (described below).

The test compression performed by coders 32 in Step 22 preferably is conducted in parallel to quickly compile data corresponding to each of the lossless coders. Parallel coding of the test data is possible due to the fact that computing power has become so inexpensive that the benefits (e.g., in terms of encoding speed) greatly outweigh the costs. Nevertheless, in an alternative embodiment, each of coders 32 shown in Fig. 1A may code the test data sequentially over a designated period of time to produce the corresponding performance data. Although not preferred, such a sequential test may be performed when computing power is at a premium.

Turning to Table 1 below, the performance criteria generated in Step 22 for nine different lossless coders (three different word lengths x three different coding techniques) is shown. Note first that the input bit rate is set at a predetermined value, while the output bit rate, although preferably set based on the transmission medium employed, may be continuously updated based on feedback information regarding the lossless coder employed. Optimally, the output bit rate will be made as small as possible. As shown in Table 1, after compressing a test amount of data, an output file size in bytes, a compression ratio, and a time to encode are each determined for a designated speed input (kbit/second) and speed output (kbit/second). For example, for an input file having 304,180,992 bytes and when using eight bits per word, Huffman coding achieves a compression ratio of 1.8272, Lempel-Ziv coding achieves a ratio of 2.505, and arithmetic coding achieves a ratio of 2.7724. In addition, the time to encode the test data for each of these algorithms is 128 seconds, 522 seconds, and 1,582 seconds, respectively. Once the performance criteria are generated for each lossless encoder, method 10 executes step 24 to select one of the coders to code, compress the data for a predetermined amount of time or for a particular amount of data.

Notably, the selection made in Step 24 typically is not based solely on compression ratio realized but rather the selection is made based on a combination of overall processing time and compression ratio performance characteristics. For example, in Table 1, arithmetic coding, for eight bits per word, achieves a compression ratio of 2.7724 which is greater than the compression ratio achieved for Lempel-ZIV coding, 2.505. However, arithmetic coding takes more than fifteen minutes longer to encode than the Lempel-Ziv lossless coder. In this case, method 10 likely would select the Lempel-Ziv coder in Step 24. However, if the performance achieved by all $n \times m$ lossless coders does not satisfy a threshold level, method 10 may decide to send the data uncompressed. This decision depends on, among other things, user requirements.

The input clock rate indicated in Table 1 is dependent upon both the media over which the data is transmitted (internet, for example) and the type of coding algorithm implemented. The time performance criteria is generated according to the following equation,

$$t_{overall} = t_c + t_{processing}. \qquad \text{(Eq. 1)}$$

In Equation 1, $t_{processing}$ includes the time duration associated with compressing the data, system delay, etc. Further, t_c is the time to transmit the data and equals the size of the file divided by the compression ratio and by the output speed, i.e., the bit rate, and reflects the time savings achieved by compressing the data. The compression ratio (CR) being equal to the input file size divided by the output file size.

Table 1

		Huffman	Lempel-Ziv	Arithmetic
8 bits per word	Output File Size (byte)	166,470,896	193,662,252	184,718,156
	Compression Ratio	1.8272	1.5707	1.6467
	Time (second)	128	265	2,742
	Input Speed (kbit/second)	19,014	9,180	887
	Output Speed (kbit/second)	10,406	5,845	538
10 bits per word	Output File Size (byte)	121,428,144	177,869,924	174,151,624
	Compression Ratio	2.505	1.7101	1.7466
	Time (second)	522	382	330
	Input Speed(kbit/second)	4,658	6,368	7,371
	Output Speed (kbit/second)	1,859	3,724	4,220
12 bits per word	Output File Size (byte)	109,716,096	142,998,504	125,961,032
	Compression Ratio	2.7724	2.1272	2.4149
	Time (second)	1,582	1,438	1,505
	Input Speed(kbit/second)	1,537	1,693	1,617
	Output Speed(kbit/second)	555	796	670

Once a coder (n,m) 32 (Fig. 1A) is selected in Step 24, method 10 encodes the data with the selected coder for, preferably, a predetermined amount of time. Thereafter, the program returns to Step 22 to code a new test portion of the data stream and select an optimum coder for encoding the next portion of the data stream. This operation may require implementing a different lossless coder shown in Chart 30.

Turning to FIG. 2, a system 40 for performing method 10 includes an encoder 41 having an input interface 42 which includes a clock input C1 and a data input D1 for receiving a data stream 43 that is either synchronous or asynchronous. Interface 42 is coupled to a digital signal processor (DSP) chip 46 via input and output data-control-synchronous input/output lines 44. Notably, DSP 46 preferably performs steps 16 and 18 in method 10 shown in FIG. 1 to frame the data and prepare it for compression. The output of DSP 46 is coupled to computer 50 via a PCI bus 48 that communicates the framed data to the computer. Computer 50, preferably, adds appropriate header codes to the data stream to indicate different packets of data and operates to encode/compress the test data with each lossless coder shown in chart 30. As noted above, computer 50 may comprise a plurality of processors each capable of encoding/compressing data for a corresponding one of the lossless coders implemented from the grid 30 in Fig. 1A. Alternatively, a single computer 50 could be used to implement the test compression for each of the lossless coders 32 in a sequential fashion for predetermined period of time.

Computer 50 may also be used to add header codes to the data to ensure that the file will be decompressed correctly. The compressed data is then transmitted via PCI bus 48 to the DSP chip 46 to divide the data as necessary for the specific communication system implemented. This process may involve buffering the data by inserting empty blocks and/or deleting existing blocks. Thereafter, particular synchronization codes may be added and the data stream is transmitted along the input/output lines back to interface 42. The particular interface code settings include designating the number of bits/word, the number of words per frame, synchronizing codes, a control sum, etc. Interface 42 then outputs the data stream on line D2, so that it may be transmitted over a medium 52 such as the internet. Preferably, the output clock rate C2 is set by the operator and is dependent upon the type of medium 52 implemented.

Next, the decoder 53 of system 40 includes an interface 54 having a data input D3 for receiving the compressed data from medium 52 at a clock rate C3 that corresponds to clock rate C2 output from interface 42. Notably, clocks C2 and C3 are optional. Interface 54 transmits the compressed data stream via data-control-synchronous lines 56, while the POC synchronization added by encoder 41 is deleted. Thereafter, a DSP chip 58 detects the header code(s) and removes empty blocks from the data stream. The data processed by DSP chip 58 is then transmitted to a computer 62 via a PCI bus 60. Computer 62 decompresses the data, and, preferably, implements conventional control sum check (CSC) comparison techniques. Additional error detection or error correction coders may also be implemented by computer 62. A Reed-Solomon error correction coder is standard for communication networks and is preferably included. Notably, the above-described processing operations may be performed either by computer(s) 50, 62 or DSP chips 46, 48, but the preferred implementation has been described. The compressed data is then transmitted back to interface 54 and onto data line D4 at a clock rate C4 = C1.

A representation of the method steps described in FIG. 1 and performed by the apparatus shown in Fig. 2 is shown schematically in FIG. 3 for a telemetric data stream. The arrow labeled A on the right-side of FIG. 3 indicates the encoding process, while the arrow B along the left-side of the data shown in FIG. 3 indicates the decoding process. More particularly, data stream 43 is input to interface 42 (FIG. 2) and then framed in a preassigned fashion into packet portions 64, 66 (preferably several kilobytes, e.g., two 8k portions), preferably by DSP 46. Thereafter, portions 64, 66 are compressed into, for example, a 4.5k block 68 and a 4.3k block 70. Then, a header 73, 75 is added to the blocks of the packets (with the interface information described above) to create blocks 72, 74, respectively. Then, the blocks are buffered to construct a buffered and compressed packet 76 which may be divided again if necessary to create stream 78. Then, POC synchronization is added by DSP chip 46 and new data stream 80 may be transmitted to decoder 53 via, for example, the internet 52 (FIG. 2) where it is decoded as described above.

Many changes and modifications may be made within the scope of the present invention without departing from the spirit thereof. Other changes and modifications falling within the scope of the invention will become apparent from the appended claims.

What is claimed is:

1. A method of lossless compression of a stream of data, the method comprising the steps of:

providing a plurality of different types of lossless coders;

selecting one of the lossless coders to compress the data stream; and

encoding the data stream with the selected lossless coder.

2. The method of claim 1, further comprising the step of, prior to said selecting step, individually compressing at least a portion of the data stream with each of the lossless coders; and

wherein said selecting step is performed based on said compressing step.

3. The method of claim 1, wherein said selecting step is based on a performance characteristic associated with said compressing step.

4. The method of claim 3, wherein the performance characteristic includes at least one of compression ratio and duration of said compressing step for a corresponding lossless coder.

5. The method of claim 1, wherein at least one of the lossless coders uses statistical modeling.

6. The method of claim 5, wherein at least another of the lossless coders uses dictionary-based modeling.

7. The method of claim 2, wherein the lossless coders perform said compressing step in parallel.

8. The method of claim 2, wherein the lossless coders perform said compressing step sequentially.

9. The method of claim 1, wherein the lossless coders are defined in part by a number of bits per word used in said encoding step.

10. A method of lossless compression of a stream of data, the method comprising the steps of:

using a plurality of different types of lossless coders to compress a test portion of the data stream;

determining a performance characteristic associated with each of the lossless coders in response to said using step;

selecting one of the lossless coders based on said determining step;

encoding a first portion of the data stream with the selected coder; and

repeating said using, determining, selecting and encoding steps for another test portion and a second portion of the data stream.

11. The method of claim 10, wherein said repeating step includes selecting a different one of the lossless coders.

12. The method of claim 10, wherein the lossless encoders perform said using step in parallel.

13. The method of claim 10, wherein the lossless encoders perform said using step sequentially.

14. The method of claim 10, wherein each of the lossless coders uses (1) a compression technique, and (2) a number of bits per word, the number being determined by said selecting step, in said encoding step.

15. The method of claim 14, wherein the compression technique is one of Arithmetic coding, Huffman coding and LZ coding.

16. The method of claim 10, wherein the data stream comprises data from a plurality of different sources.

17. The method of claim 10, wherein the performance characteristic includes at least one of compression ratio and duration of said using step for a corresponding lossless coder.

18. An apparatus for lossless data compression, the apparatus comprising:

an interface to receive a stream of data;

a plurality of different types of lossless coders;

a processor; and

wherein each said lossless coder separately compresses a test portion of the data stream and, in response, said processor (1) determines a performance characteristic associated with each said lossless coder, and (2) selects, based on said performance characteristics, one of said lossless coders to encode at least a first portion of the data stream.

19. The method of claim 18, wherein the performance characteristic includes at least one of compression ratio and duration of the compression of the test portion.

20. The apparatus of claim 18, wherein said encoder includes a plurality of processors and each said lossless coder corresponds to one of the processors, and wherein said lossless coders compress the same test portion in parallel.

21. A method of claim 18, wherein the data stream comprises data from a plurality of different sources.

22. An apparatus for lossless data compression, the apparatus comprising:

an encoder including an interface to receive a stream of data, a plurality of different types of lossless coders and a processor, wherein each said lossless coder separately compresses a test portion of the data stream and, in response, said processor (1) determines a performance characteristic associated with each said lossless coder, and (2) selects, based on said performance characteristics, one of said lossless coders to encode at least a first portion of the data stream; and

a decoder that receives and decompresses said encoded first portion of the data stream.

23. A method of lossless compression of a stream of data, the method comprising the steps of:

using a plurality of different types of lossless coders to compress a test portion of the data stream;

determining a performance characteristic associated with each of the lossless coders in response to said using step;

selecting one of the lossless coders based on said determining step;

encoding a first portion of the data stream with the selected coder; and

repeating said using, determining, selecting and encoding steps for another test portion and a second portion of the data stream, wherein said repeating step includes selecting a different one of the lossless coders;

transmitting the encoded first portion via a communication medium to a decoder; and

decompressing the encoded first portion.

24. The apparatus of claim 23, wherein the medium is the internet.

25. The apparatus of claim 23, wherein the performance characteristic includes at least one of compression ratio and duration of said using step for corresponding lossless coder.

26. The apparatus of claim 23, wherein the data stream comprises data from a plurality of different sources.

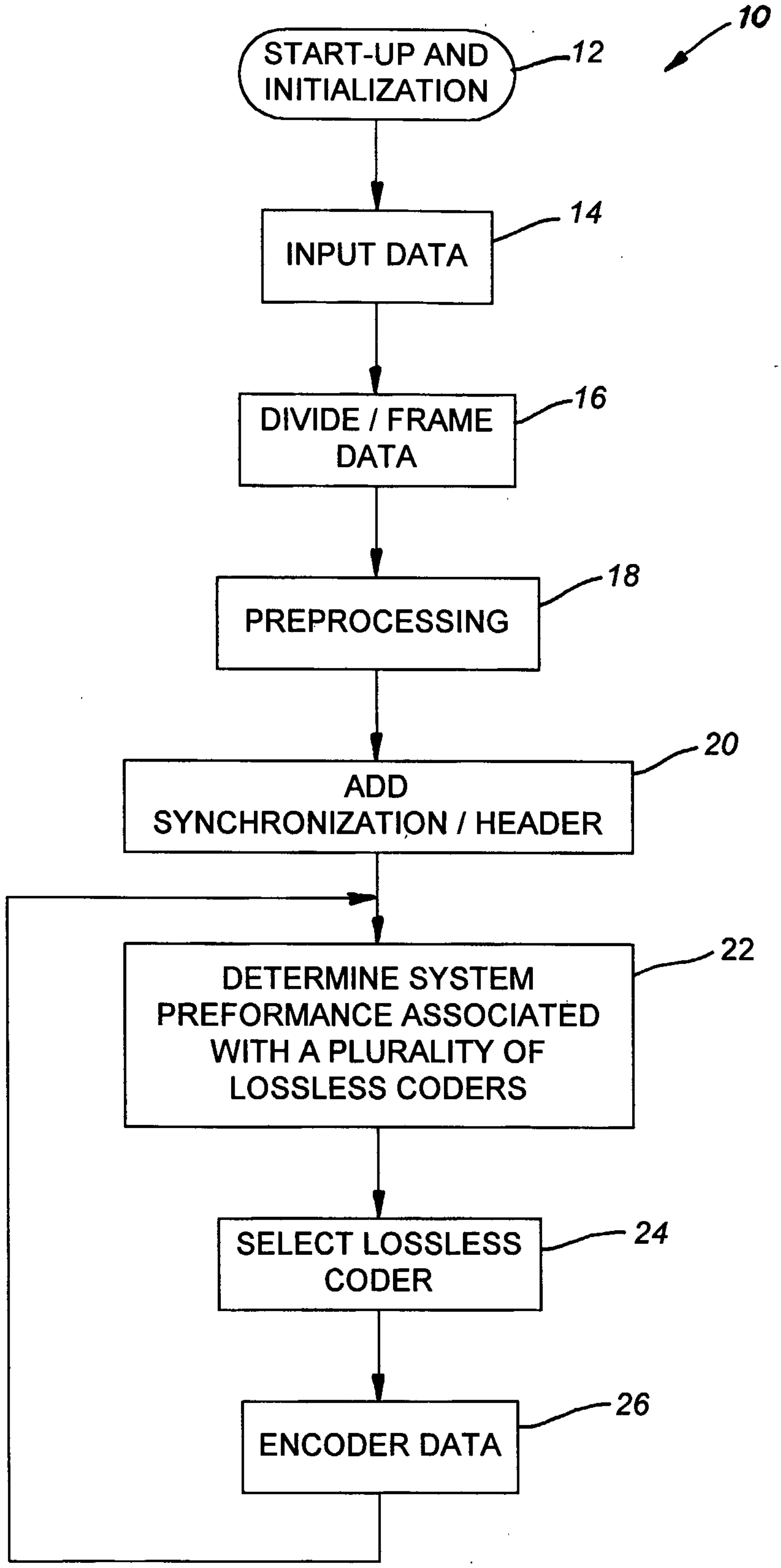
10
START-UP AND INITIALIZATION
12
INPUT DATA
14
DIVIDE / FRAME DATA
16
PREPROCESSING
18
ADD SYNCHRONIZATION / HEADER
20
DETERMINE SYSTEM PREFORMANCE ASSOCIATED WITH A PLURALITY OF LOSSLESS CODERS
22
SELECT LOSSLESS CODER
24
ENCODER DATA
26

FIG. 1

30

32

	ARITHMETIC	LEMPEL-ZIV	HUFFMAN	...	n
8	LOSSLESS CODER (Ar,8)	LOSSLESS CODER (LZ,8)			LOSSLESS CODER (n,8)
10	LOSSLESS CODER (Ar,10)				
...					
m	LOSSLESS CODER (Ar,M)				LOSSLESS CODER (n,m)

FIG. 1A

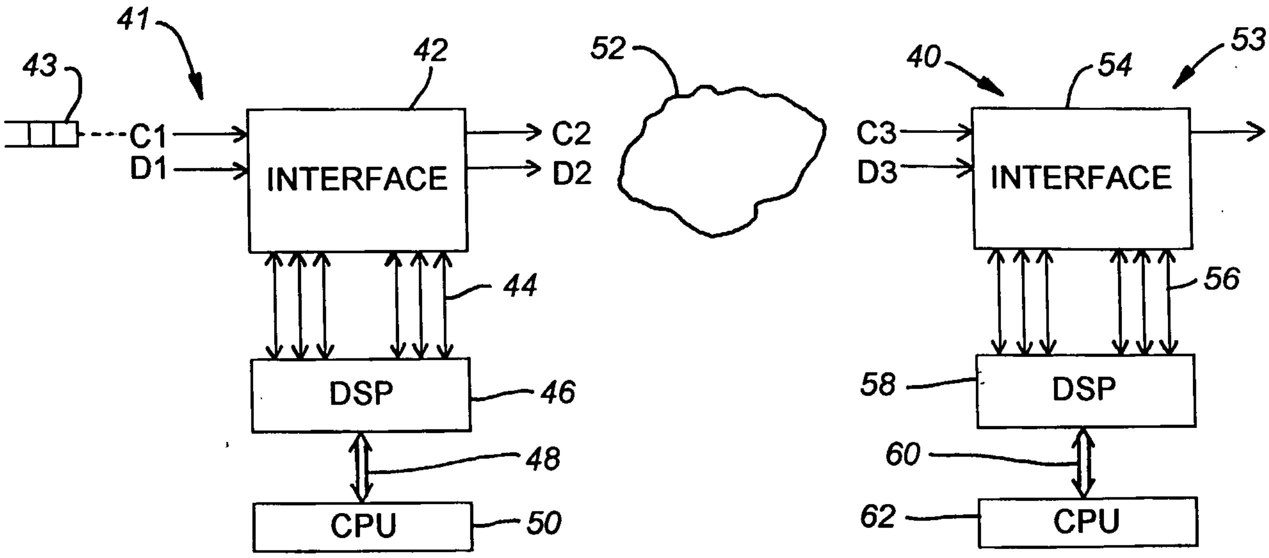
41
43
42
52
40
54
53
C1
D1
INTERFACE
C2
D2
C3
D3
INTERFACE
44
56
DSP
46
58
DSP
48
60
CPU
50
62
CPU

FIG. 2

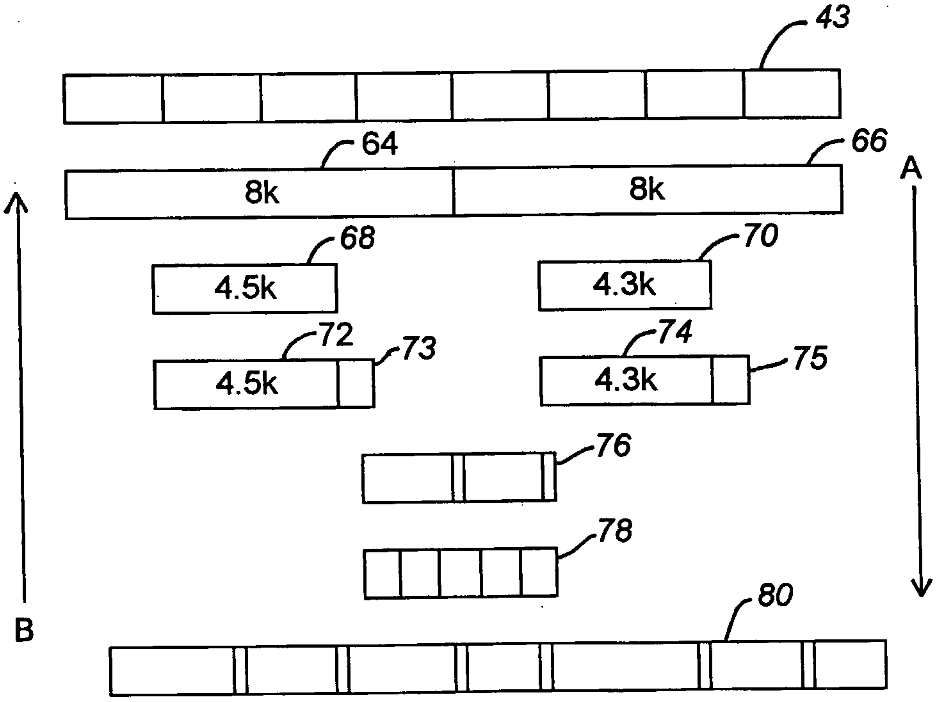
43
64
66
8k
8k
A
68
4.5k
70
4.3k
72
73
4.5k
74
4.3k
75
76
78
80
B

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/05722

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H03M 7/34; H04N 1/415
US CL : 341/51, 67; 358/426
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 341/51, 67; 358/426

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST, EAST
Search terms: lossless, cod$3, switch$, multiplex$, select$, compress$

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,708,511 A [GANDHI et al.] 13 January 1998, (13/01/98) figs. 1, 4, 7, 8; abstract, col 2, lines 32-37; col 5, lines 33 - 36; col 6, lines 7 - 9, 55 - 59; col 7, lines 22 - 25.	1 - 16 --------- 17 - 261 -
Y	US 5,485,526 A [TOBIN] 16 January 1996, (16/01/96) fig. 1.	17 - 26

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier document published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"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)	"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
"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	"&" document member of the same patent family

Date of the actual completion of the international search 29 MARCH 2001	Date of mailing of the international search report 27 APR 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer JEAN JEANGLAUDE Telephone No. 703 - 306-3405

Form PCT/ISA/210 (second sheet) (July 1998)★