(12) UK Patent Application (19) GB (11) 2539239 (13) A

(43) Date of A Publication 14.12.2016

(21) Application No: 1510127.2

(22) Date of Filing: 10.06.2015

(71) Applicant(s):
Gurulogic Microsystems Oy
Linnankatu 34, Turku 20100, Finland

(72) Inventor(s):
Ossi Kalevo

(74) Agent and/or Address for Service:
Basck Ltd
16 Saxon Road, CAMBRIDGE, Cambridgeshire,
CB5 8HS, United Kingdom

(51) INT CL:
H04L 29/06 (2006.01) ***H03M 7/48*** (2006.01)

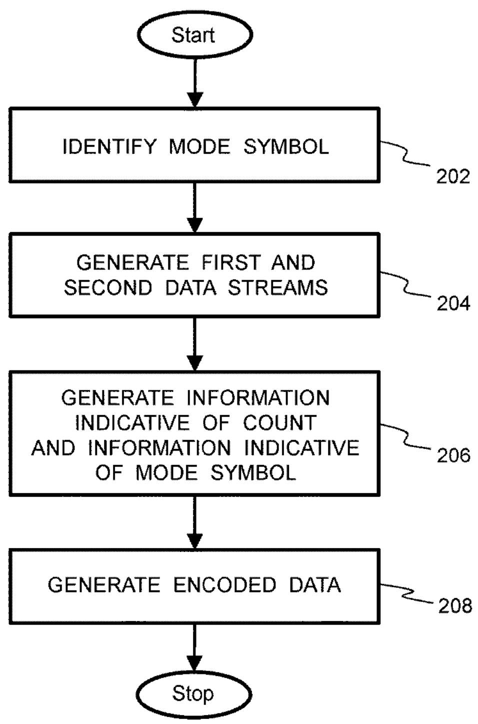
(56) Documents Cited:
WO 2010/115789 A1 **WO 2005/043921 A1**
CN 001949670 A

(58) Field of Search:
INT CL **G06F, H03M, H04L, H04N**
Other: **WPI, EPODOC, INSPEC**

(54) Title of the Invention: **Encoders, decoders and methods**
Abstract Title: **Run-length encoding (RLE) of data symbols including identifying the most frequent (mode) symbol**

(57) An encoder is operable to analyze input data to identify at least one most frequently-occurring (mode) symbol therein. The encoder generates a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol. The encoder also generates information indicative of a count of the non-mode symbols in the first data stream and information indicative of the at least one mode symbol. The encoder then encodes the information indicative of the at least one mode symbol and the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate encoded data.

FIG. 2



GB 2539239 A

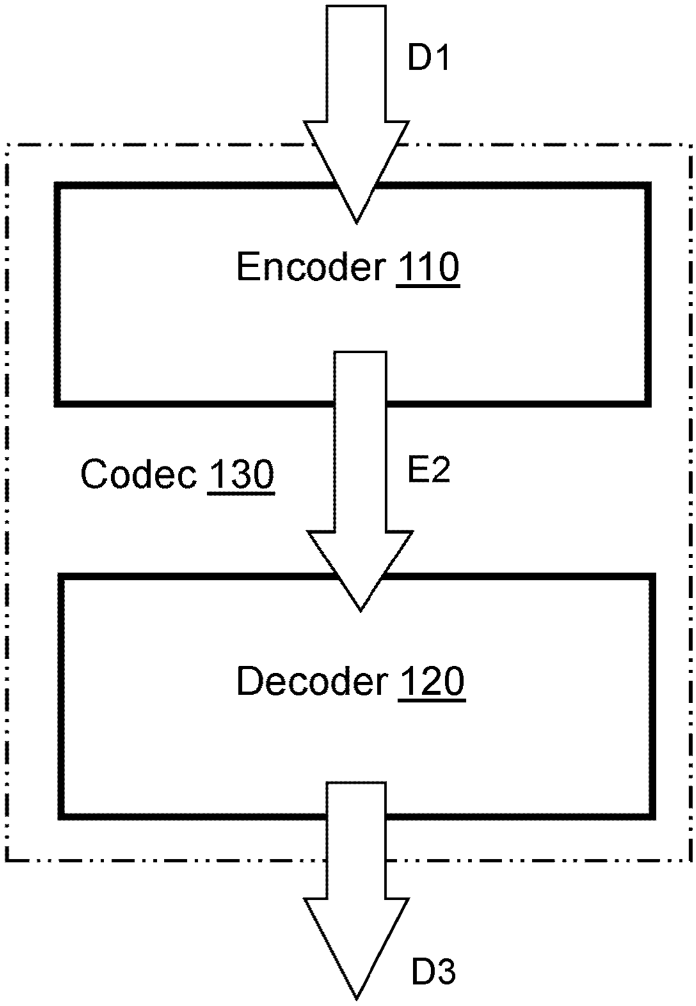


FIG. 1

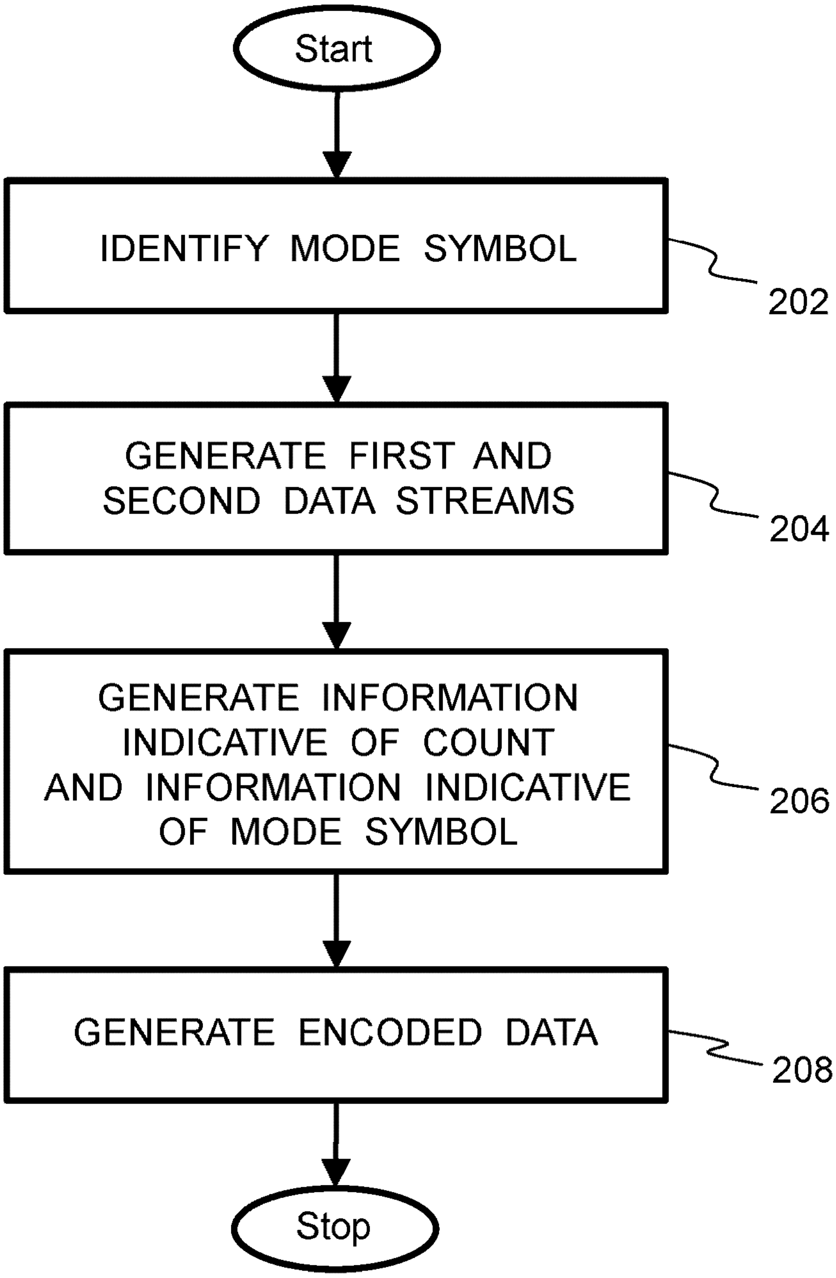

Start
IDENTIFY MODE SYMBOL
202
GENERATE FIRST AND SECOND DATA STREAMS
204
GENERATE INFORMATION INDICATIVE OF COUNT AND INFORMATION INDICATIVE OF MODE SYMBOL
206
GENERATE ENCODED DATA
208
Stop

FIG. 2

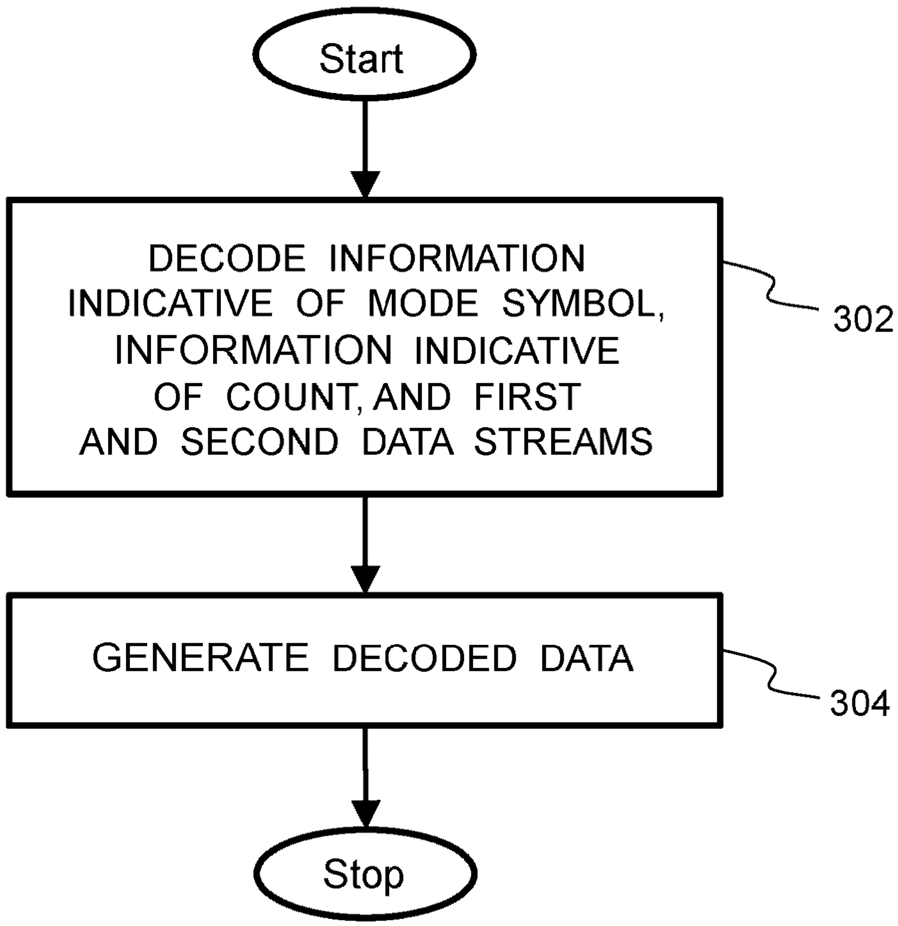

Start
DECODE INFORMATION INDICATIVE OF MODE SYMBOL, INFORMATION INDICATIVE OF COUNT, AND FIRST AND SECOND DATA STREAMS
302
GENERATE DECODED DATA
304
Stop

FIG. 3

ENCODERS, DECODERS AND METHODS

TECHNICAL FIELD

The present disclosure relates to encoders for encoding input data (D1) to generate corresponding encoded data (E2), and corresponding methods of encoding input data (D1) to generate corresponding encoded data (E2). Moreover, the present disclosure relates to decoders for decoding encoded data (E2) to generate corresponding decoded data (D3), and corresponding methods of decoding encoded data (E2) to generate corresponding decoded data (D3). Furthermore, the present disclosure is concerned with computer program products comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute aforesaid methods. Additionally, the present disclosure concerns codecs including at least one aforementioned encoder and at least one aforementioned decoder.

BACKGROUND

Run Length Encoding (RLE) is an encoding method that can be used to compress data by reducing an associated count of symbols, which is achieved by separately delivering consecutive runs of the symbols (see reference [1]). Moreover, Split RLE or SRLE (see reference [2]) is a version of RLE that can be used to deliver and compress the symbols and the consecutive runs of the symbols separately.

There are multiple different implementations of RLE that are contemporarily available. As an example, in some implementations, consecutive runs can be delivered for all different symbols. In other implementations, the consecutive runs are delivered only when there are at least two similar symbols in a symbol stream.

Furthermore, ZRLE (see reference [3]) is one special version of RLE that separately delivers non-zero values, and runs of zero values before non-zero values. ZRLE is an efficient method when given data contains a lot of zero values. ZRLE is used, for example, when quantized Discrete Cosine Transform (DCT) coefficients are compressed in Joint Photographic Experts Group (JPEG) compression.

The aforementioned RLE and SRLE can be used for bits, numbers or symbols, whereas ZRLE can be used only for numbers.

RLE and SRLE are efficient methods only in a case where there are a lot of consecutive symbols in the data, but none of those symbols is a dominant symbol. ZRLE is an efficient method only when “zero” is a dominant value in the data.

Unfortunately, when JPEG uses the ZRLE method, zero is not always the dominant value, especially when a desired quality is high and a quantization rate is low. This makes the ZRLE method very inefficient to use in such circumstances.

Therefore, there exists a need for alternative encoding methods that address shortcomings associated with known methods of encoding data, for example as aforementioned. These shortcomings relate, for example, to a degree of data compression that is achievable.

SUMMARY

The present disclosure seeks to provide an improved encoder for encoding input data (D1) to generate corresponding encoded data (E2).

Moreover, the present disclosure seeks to provide an improved decoder for decoding encoded data (E2) to generate corresponding decoded data (D3).

A further aim of the present disclosure is to at least partially overcome at least some of the problems of the prior art, as discussed above.

In a first aspect, embodiments of the present disclosure provide an encoder for encoding input data (D1) to generate corresponding encoded data (E2), characterized in that:

(a) the encoder is operable to analyze the input data (D1) to identify at least one mode symbol therein;

(b) the encoder is operable to generate a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) the encoder is operable to generate information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) the encoder is operable to encode the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

The aforementioned encoder can be used for different types of symbols, for example, including numbers, bits, alphanumeric characters, words, and so forth.

The aforementioned encoder is capable of achieving a higher compression ratio for generating lossless compressed data as compared to other known conventional compression methods, when the input data (D1) includes at least one dominant symbol, namely the at least one mode symbol. Moreover, the encoded data (E2) so generated is susceptible to being decoded rapidly, as compared to conventional entropy coding methods.

In a second aspect, embodiments of the present disclosure provide a method of encoding input data (D1) to generate corresponding encoded data (E2), characterized in that the method includes:

(a) analyzing the input data (D1) to identify at least one mode symbol therein;

(b) generating a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) generating information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) encoding the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

In a third aspect, embodiments of the present disclosure provide a decoder for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

(i) the decoder is operable to decode from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) the decoder is operable to insert before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

In a fourth aspect, embodiments of the present disclosure provide a method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

(i) decoding from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) inserting before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

In a fifth aspect, embodiments of the present disclosure provide a codec including at least one encoder for encoding input data (D1) to generate corresponding encoded data (E2) pursuant to embodiments of the present disclosure, and at least one decoder for decoding the encoded data (E2) to generate corresponding decoded data (D3) pursuant to embodiments of the present disclosure.

In a sixth aspect, embodiments of the present disclosure provide a computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute any of the aforementioned methods pursuant to embodiments of the present disclosure.

Additional aspects, advantages, features and objects of the present disclosure would be made apparent from the drawings and the detailed description of the illustrative embodiments construed in conjunction with the appended claims that follow.

It will be appreciated that features of the present disclosure are susceptible to being combined in various combinations without departing from the scope of the present disclosure as defined by the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The summary above, as well as the following detailed description of illustrative embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the present disclosure, exemplary constructions of the disclosure are shown in the drawings. However, the present disclosure is not limited to specific methods and apparatus disclosed herein. Moreover, those in the art will understand that the drawings are not to scale. Wherever possible, like elements have been indicated by identical numbers.

Embodiments of the present disclosure will now be described, by way of example only, with reference to the following diagrams wherein:

FIG. 1 is a schematic illustration of an encoder for encoding input data (D1) to generate corresponding encoded data (E2) and a decoder for decoding the encoded data (E2) to generate corresponding decoded data (D3), wherein the encoder and the decoder collectively form a codec, in accordance with an embodiment of the present disclosure;

FIG. 2 is a schematic illustration of a flow chart depicting steps of a method of encoding input data (D1) to generate corresponding encoded data (E2), in accordance with an embodiment of the present disclosure; and

FIG. 3 is a schematic illustration of a flow chart depicting steps of a method of decoding encoded data (E2) to generate corresponding decoded data (D3), in accordance with an embodiment of the present disclosure.

In the accompanying diagrams, an underlined number is employed to represent an item over which the underlined number is positioned or an item to which the underlined number is adjacent. A non-underlined number relates to an item identified by a line linking the non-underlined number to the item.

DETAILED DESCRIPTION OF EMBODIMENTS

The following detailed description illustrates embodiments of the present disclosure and ways in which they can be implemented. Although some modes of carrying out the present disclosure have been disclosed, those skilled in the art would recognize that other embodiments for carrying out or practicing the present disclosure are also possible.

In a first aspect, embodiments of the present disclosure provide an encoder for encoding input data (D1) to generate corresponding encoded data (E2), characterized in that:

(a) the encoder is operable to analyze the input data (D1) to identify at least one mode symbol therein;

(b) the encoder is operable to generate a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) the encoder is operable to generate information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) the encoder is operable to encode the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

The input data (D1) can be any sort of data that can be represented with symbols. Examples of symbols include, but are not limited to, alphabetical characters, alphanumeric characters, numeric characters, and sequences thereof. Moreover, the input data (D1) can be original data, source data, data processed with some other encoding method, or a stream of data. Examples of the input data (D1) include, but are not limited to, binary data, text data, audio data, image data, video data, measurement data, sensor data, biometric data, biological genetic data, genome data, transform coefficient data, transformed data, processed data, and partial data.

With reference to the aforementioned transform coefficient data and the aforementioned transformed data, some examples of associated transformations in respect of the input data (D1) concerned, coefficients of which are capable of being employed, in respect of the aforementioned method or methods pursuant to the present disclosure include:

DCT (Discrete Cosine Transform), DFT/FFT (Discrete/Fast Fourier Transform), Hadamard transform, Haar transform, wavelets, DST (Discrete Sine Transform), KLT (Karhunen-Loeve Transform), linear transformations, affine transformations, reflections, translations, rotations, scaling, shear, multilevel coding, ODelta coding, quantization, color space transformations, linear filters (FIR, IIR), non-linear transforms (partial functions) and non-linear filters (median, mode), but not limited thereto. Such aforementioned transformations are optionally used to process the input data (D1) prior to corresponding processed data being employed by the method or methods pursuant to the present disclosure.

The aforementioned encoder is very efficient when there is at least one dominant symbol in the input data (D1). Throughout the present disclosure, the term "*mode symbol*" refers to a symbol that is dominant in the input data (D1), while the term "*non-mode symbol*" refers to a symbol that is not dominant in the input data (D1). In other words, a mode symbol is a symbol that occurs most often in the input data (D1), while a non-mode symbol is a symbol other than the mode symbol.

It will be appreciated that "*mode*" is not same as "*zero*", although mode symbols can sometimes be equal to zero in value, if zero is a most dominant value in given input data (D1).

It is to be noted here that there can be more than one mode symbol in the input data (D1), and embodiments of the present disclosure can be implemented in a similar manner for a case where there are more than one mode symbol. Optionally, in such a case, the encoder is operable to recursively process (a) to (d) for each mode symbol.

According to an embodiment, the encoder is operable to generate symbols in the input data (D1) prior to (a) to (d) by splitting or combining original symbols present in the input data (D1). Optionally, in this regard, the encoder is operable:

(i) to split at least one or more portions of the input data (D1) to generate one or more new symbols for processing pursuant to (a) to (d); and/or

(ii) to combine one or more alphanumeric characters of at least one or more portions of the input data (D1) to generate new combined symbols for processing pursuant to (a) to (d).

In order to illustrate how the input data (D1) can be split in (i), there will now be considered an example where an input data stream (D1) is a bit stream including 24 bits as follows:

1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1

The bit stream can be split into N-bit symbols that can then be compressed pursuant to (a) to (d). In the example herein, portions of the bit stream can be split to generate new 4-bit symbols as follows:

(1, 0, 0, 0), (0, 1, 1, 1), (1, 0, 1, 1), (1, 0, 0, 0), (1, 0, 0, 0), (1, 0, 1, 1)

Hereinabove, the new 4-bit symbols are separated by brackets, for the sake of clarity only.

Next, in order to illustrate how alphanumeric characters can be combined in (ii), there will now be considered an example where an input data stream (D1) includes 20 alphanumeric characters as follows:

A, B, B, A, A, A, B, B, A, B, A, B, A, B, B, A, B, B, A, A

In the example herein, certain sequences of alphanumeric characters, namely ‘A, B, B’, ‘A, A’ and ‘A, B’ occur repeatedly. Therefore, these alphanumeric characters can be combined to generate new combined symbols as follows:

ABB, AA, ABB, AB, AB, ABB, ABB, AA

The processing pursuant to (a) to (d) can then be applied on the new symbols.

Moreover, according to an embodiment, the encoder is operable to process recursively the first data stream including the non-mode symbols pursuant to (a) to (d). An example of how the recursive processing can be performed is provided later.

According to an embodiment, the encoder is operable to interleave data of the first and second data streams to provide a resultant data stream for compression in (d). According to another embodiment, the encoder is operable to interleave data of the first and second data streams as interleaved data blocks to provide a resultant data stream for compression in (d). Examples of how the data of the first and second data streams can be interleaved will be provided later.

Optionally, the information indicative of the count of the non-mode symbols in the first stream is delivered before the first and second data streams in the encoded data (E2).

According to an embodiment, the encoder is operable to deliver the encoded data (E2) to at least one decoder. As an example, the encoded data (E2) is written to a file by the encoder, and is then read from the file by the at least one decoder. As another example, the encoded data (E2) is streamed from the encoder to the at least one decoder.

According to an embodiment, the encoder is operable to deliver, to the at least one decoder, at least one index to indicate whether or not the at least one mode symbol is communicated in the encoded data (E2). Optionally, the at least one index is used to indicate how many mode symbols are included, namely a count of mode symbols.

Moreover, optionally, the at least one index that is provided to the at least one decoder contains also information about an encoding method employed by the encoder, for example, such as a Mode RLE (MRLE) or a Split mode RLE (SMRLE) method. In this regard, the at least one index can be referred to as a method selection index. An example of such a method selection index is 'SMRLE2_X_X', wherein a number '2' in the index indicates that there are two mode symbols in the encoded data (E2), and each symbol 'X' in the index indicates that the mode symbols are delivered within the encoded data (E2). As another example, when a count of mode symbols is delivered within the encoded data (E2), a method selection index 'SMRLEX' can be used, wherein the symbol 'X' indicates that the count of mode symbols and the mode symbols are delivered within the encoded data (E2). As yet another example, when the mode symbols are not delivered within the encoded data (E2), a method selection index 'SMRLE2_H_F' can be used, wherein symbols 'H' and 'F' represent mode symbols that are used. As still another example, when one mode symbol is used and is delivered within the encoded data (E2), a method selection index 'SMRLE1_X' can be used.

In this manner, the at least one mode symbol and a count of mode symbols are delivered to the at least one decoder either within the encoded data (E2) or via the at least one index.

Moreover, optionally, the at least one index can be used to indicate whether or not a first symbol, a last symbol or both the first and last symbols in the input data (D1) are non-mode symbols. As an example, a method selection index 'SMRLE1_X_First' indicates that the first symbol in the input data (D1) is a non-mode symbol, while a method selection index 'SMRLE1_X_Last' indicates that the last symbol in the input data (D1) is a non-mode symbol. Likewise, a method selection index 'SMRLE1_X_FirstAndLast' indicates that both of the first and last symbols in the input data (D1) are non-mode symbols.

According to an embodiment, the encoder is operable to encode the information that is indicative of the at least one mode symbol to generate the encoded data (E2) at least in part as prior information to the at least one decoder. In this embodiment, the at least one mode symbol and the count of mode symbols are delivered as prior information before remainder of the encoded data (E2) is delivered.

According to an embodiment, the encoder is operable to deliver additional information, in the encoded data (E2), relating to delivery of at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, and/or maximum indices of different data. Delivery of the probability tables and indices has been described in detail in the patent application GB1403039.9, filed by Gurulogic Microsystems Oy.

According to an embodiment, the encoder is operable to dynamically select, from amongst a plurality of encoding algorithms, an encoding algorithm to be employed in (a) to (d), depending upon one or more properties of the input data (D1), to obtain an improved degree of data compression in the encoded data (E2).

In this regard, optionally, a most efficient encoding algorithm is selected in a dynamically changing manner, as a nature or content of the input data (D1) to be encoded varies; "*nature*" in such case refers to a structure of the input data (D1), for example a spectrum of value distributions within the input data (D1), localized concentrations of certain numerical values within the input data (D1), patterns of data repetition within the input data (D1), but not limited thereto. As mentioned earlier, examples of the input data (D1) include, but are not limited to, binary data, text data, audio data, image data, video data, measurement data, sensor data, biometric data, biological genetic data, genome data, transform coefficient data, transformed data, processed data, partial data.

Optionally, the selection of the most efficient encoding algorithm is performed by:

(i) compressing the input data (D1) with different encoding algorithms;

(ii) computing an entropy of compressed data obtained with each of the different encoding algorithms;

(iii) computing, for each of the different encoding algorithms, an amount of data required for delivery of the aforementioned additional information relating to at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, and/or maximum indices of different data; and

(iv) comparing the different encoding algorithms with respect to the entropy of the compressed data and/or the amount of data required for delivery of the additional information.

As an example, some portions of the input data (D1) can be compressed by range coding those portions of the input data (D1) directly, if delivery of a range table can be executed efficiently.

According to an embodiment, the encoder is operable to employ range encoding data compression and/or Huffman encoding data compression to compress the second data stream when generating the encoded data (E2).

According to an embodiment, the encoder is operable to employ additional encoding methods for encoding one or more portions of the input data (D1), wherein the additional methods include at least one of: Delta encoding, ODelta encoding, Run Length Encoding (RLE), Split RLE (SRLE), ZRLE, Huffman coding, Range coding, Entropy Modifying (EM) encoding, Continuum operator encoding. Herein, the term "*Delta encoding*" refers to a method of storing or transmitting data in a form of differences between sequential data rather than complete data files, while the term "*ODelta*" refers to a differential form of encoding based upon wraparound in a binary counting regime, for example as described in the patent document GB 1412937.3. RLE, SRLE and ZRLE can be employed, for example, as described in references [1], [2] and [3], respectively.

The Continuum operator encoding can be employed, for example as described in reference [4]. The EM encoding can be employed, for example, as described in reference [5].

Optionally, the first data stream and the second data stream are compressed separately, by using different entropy encoding methods.

As an example, a method selection index ‘ODeltaSMRLE1_XCoded’ indicates that the input data (D1) is first ODelta-coded, then encoded using one or more encoding methods pursuant to embodiments of the present disclosure; in this example, SMRLE, with one delivered mode symbol, is employed to generate the mode symbol and first and second data streams, and then the first and the second data streams are encoded using one or more predefined (or selected with delivery of selection) entropy encoding methods, such as Range coding, Huffman coding or Copy coding. Encoded data (E2) generated in such a manner is decoded at the at least one decoder in a reverse order of processing steps to an order of processing steps employed during encoding of the input data (D1) to generate the encoded data (E2); such reverse order is employed when transcoding is not required in the at least one decoder.

Furthermore, optionally, the aforementioned encoder is useable with other known encoders; for example, in conjunction with a block encoder as described in reference [6].

For illustration purposes only, there will next be considered some examples of how the aforementioned encoder can be implemented pursuant to embodiments of the present disclosure. There will now be considered a portion of an input data stream (D1) that includes 38 symbols (alphanumeric characters), represented as follows:

FHFFFJFFFFHFFFFHFHFFFIFFFFFFHFHIFFFFHF

In the portion of the input data stream (D1), counts of different symbols are as follows:

F (28), **H** (7), **I** (2), **J** (1)

As the count of the symbol '**F**' is higher than the counts of other symbols, the symbol '**F**' is determined to be a mode symbol in the input data stream (D1).

It will be appreciated that the "*counts*" mentioned here are actually values in a probability table. Since in this example, only a few different symbols occur, then it is potentially advantageous to deliver an index table for the symbols and their probabilities, or rather frequencies of occurrence thereof (namely "counts") than to deliver the entire probability table. In this example, most of other possible different symbols have a zero frequency of occurrence. Thus, indices of the symbols '**F**', '**H**', '**I**', and '**J**' are, for example, delivered one after the other, namely as an index table. As an example, the symbols '**F**', '**H**', '**I**', and '**J**' can correspond in coding to the indices '0', '1', '2' and '3', respectively, after which their probabilities '28', '7', '2' and '1', namely the probability table, is optionally sent.

In the example here, on account of the symbols of the input data (D1) being alphanumeric characters, and not integers, the index of 'F' is optionally calculated, for example, as follows:

index (F) = ascii code (F) - ascii code (A),

namely 70 – 65 = 5.

It will also be appreciated that, as the input data (D1) is encoded with the aforementioned MRLE/SMRLE method, there is an option of selecting whether used mode symbols are sent as separate values or are obtained directly from sent probability tables and index tables. If the used mode symbols are sent as separate values, then they are optionally no longer sent in connection with the sent tables. If the used mode symbols are obtained directly from the sent probability tables and index tables, then there is no requirement for the used mode symbols to be sent separately any more.

In accordance with an embodiment of the present disclosure, the encoder is operable to deliver the mode symbol, a non-mode stream, namely a first stream that includes non-mode symbols as they occur in the portion of the input data stream (D1), and a runs stream, namely a second stream that includes runs of the mode symbol occurring before each of the non-mode symbols.

Thus, the non-mode stream (namely, the first stream) can be represented as follows:

HJHHHIHHIH

, while the runs stream (namely, the second stream) can be represented as follows:

1, 3, 4, 4, 1, 3, 6, 1, 0, 4, (1)

In the illustrated example, there is one occurrence of the mode symbol (hereinafter referred to as '*the last run of the mode symbol*') after the last non-mode symbol in the portion of the input data stream (D1). Optionally, the last run of the mode symbol after the last non-mode symbol can be described by delivering a total count of symbols, which is 38 in the illustrated example. Alternatively, optionally, the last run of the mode symbol after the last non-mode symbol can be described by adding a value of '1' to an end of the runs stream, as shown in brackets above.

When the value of '1' is not added, the count of the runs of the mode symbol in the runs stream is the same as the count of the non-mode symbols in the non-mode stream, which is 10 in the illustrated example. In such a case, delivering the total count of symbols with encoded data (E2) enables determination of the last run of the mode symbol after the last non-mode symbol during subsequent decoding of the encoded data (E2).

On the other hand, when the value of '1' is added, the count of the runs of the mode symbol in the runs stream is one more than the count of the non-mode symbols in the non-mode stream. This is indicative of a presence of the last run of the mode symbol after the last non-mode symbol.

Optionally, a continuum operator, for example as described in reference [4], can be employed by the encoder pursuant to embodiments of the present disclosure. If the continuum operator is employed, the count of the runs of the mode symbol can be increased, as the count of the non-mode symbols in the non-mode stream remains the same. The original runs of the mode symbol are later obtained after decoding the continuum operator at a given decoder.

In the aforementioned example, the delivery is for "*information that is indicative of the count of the non-mode symbols*", instead of the delivery of "*the count of the non-mode symbols*". However, it will be appreciated that embodiments of the present invention encompass both approaches when coding to generate the encoded data (E2).

In a first example, the mode symbol, the non-mode stream (namely, the first stream) and the runs stream (namely, the second stream) can be delivered as interleaved data blocks one after another, as represented below:

F, 10, H, J, H, H, H, I, H, H, I, H, 1, 3, 4, 4, 1, 3, 6, 1, 0, 4, 1

Hereinabove, a first symbol 'F' indicates the mode symbol, and a second symbol '10' indicates the count of the non-mode symbols. Delivering the count of the non-mode symbols enables the given decoder to decode the encoded data (E2) properly, by using the count of the non-mode symbols as a limiter of a counter in a decoding loop employed in the given decoder. As aforementioned, it will be appreciated that the delivery of the count of the non-mode symbols can also be replaced by delivering the total count of symbols, and still the given decoder is able to operate correctly, without having been delivered a particular value of the count of the non-mode symbols, because the given decoder received other relevant values and obtained, from the received values, information it needed, namely the count of the non-mode symbols. Thus, the information about the "*count*" is optionally obtained in other ways than via only delivering that particular value.

It will be appreciated that embodiments of the present disclosure enable delivery of the information about this "*count*", in such a way that the given decoder is able to finish at a correct place. Optionally, it is irrelevant which "*count*", or a combination of "*counts*", is delivered, as long as the limiter of the counter for the decoding loop employed in the given decoder can be computed correctly.

Moreover, the runs of the mode symbol are delivered after the last non-mode symbol. Therefore, the total count of symbols is not required to be delivered. If the total count of symbols is required to be known, it can be calculated from the count of the non-mode symbols and the runs of the mode symbol, namely by summing them all together. It will be appreciated that any of the two counts, namely the total count of symbols and the count of the non-mode symbols, can be calculated, if the other is known. Alternatively, optionally, as a minimum requirement, one of these two counts is known and the first and the second streams are split for purposes of decoding, for example with the help of interleaving, in such a way that the decoding can be completed with the help of the known count.

In the first example, a method selection index 'SMRLE1_X' is delivered from the encoder to the given decoder. The method selection index 'SMRLE1_X' indicates that there is one mode symbol, and the mode symbol has been provided in the encoded data (E2).

In the aforementioned first example, only 23 symbols are required to be delivered from the encoder to the given decoder.

In a second example, the non-mode symbols and the runs of the mode symbol occurring before, after and between the non-mode symbols can be delivered as represented below:

F, 10, 1, H, 3, J, 4, H, 4, H, 1, H, 3, I, 6, H, 1, H, 0, I, 4, H, 1

The non-mode symbols and the runs of the mode symbol are interleaved in the above manner, for example, when it is not beneficial to compress the non-mode stream and the runs stream separately.

The encoded data (E2) as delivered in the second example is relatively easy to understand and to decode. As earlier, a first symbol 'F' indicates the mode symbol, and a second symbol '10' indicates the count of the non-mode symbols. During decoding, all the runs of the mode symbol (namely, numbers) are replaced with a similar number of occurrences of the mode symbol. Consequently, decoded data (D3) is then provided as follows:

FHFFFJFFFFHFFFFHFHFFFIFFFFFFHFHIFFFFHF

Furthermore, in the above example, it can be seen that the non-mode stream includes multiple repetitions of the symbol 'H'. Optionally, in this regard, a recursive processing of the non-mode stream can be performed pursuant to (a) to (d) to deliver another stream, which can be represented as follows:

H, 3, 1, J, 3, I, 2, I, 1

Hereinabove, a symbol 'H' indicates a second mode symbol, and a symbol '3' indicates the count of remaining non-mode symbols. Thereafter, the remaining non-mode symbols and runs of the second mode symbol occurring before, after and between the remaining non-mode symbols are delivered as shown above.

In a third example, the entire encoded data (E2) can be delivered as follows:

H, F, 3, 10, J, I, I, 1, 3, 2, 1, 1, 3, 4, 4, 1, 3, 6, 1, 0, 4, 1

In the third example, a method selection index 'SMRLE2_X_X' is delivered from the encoder to the given decoder. The method selection index 'SMRLE2_X_X' indicates that there are two mode symbols (namely, the symbols 'H' and 'F'), and these mode symbols have been provided in the encoded data (E2).

In a fourth example, the count of the mode symbols can be delivered within the encoded data (E2), which can be represented as follows:

2,   H, F,   3, 10,   J, I, I,   1, 3, 2, 1,   1, 3, 4, 4, 1, 3, 6, 1, 0, 4, 1

In the fourth example, a method selection index ‘SMRLEX’ is delivered from the encoder to the given decoder. The method selection index ‘SMRLEX’ indicates that the mode symbols and the count of the mode symbols have been provided in the encoded data (E2).

In a fifth example, the mode symbols and the count of the mode symbols are not delivered within the encoded data (E2), which can be represented as follows:

3, 10,   J, I, I,   1, 3, 2, 1,   1, 3, 4, 4, 1, 3, 6, 1, 0, 4, 1

In the fifth example, a method selection index ‘SMRLE2_H_F’ is delivered from the encoder to the given decoder. The method selection index ‘SMRLE2_H_F’ indicates that there are two mode symbols, namely the mode symbols ‘H’ and ‘F’.

Furthermore, optionally, the streams of runs for all mode symbols can be combined and compressed together.

In a sixth example, a combined runs stream can be represented as follows:

1, 3, 2, 1, 1, 3, 4, 4, 1, 3, 6, 1, 0, 4, 1

In the combined runs stream, counts of different run symbols are as follows:

0 (1), **1** (6), **2** (1), **3** (3), **4** (3), **6** (1)

As an example, the combined runs stream can be compressed by using range coding or Huffman coding methods.

For illustration purposes only, there will next be considered another example of how the aforementioned encoder can be implemented pursuant to embodiments of the present disclosure. There will now be considered a portion of an input data stream that includes 12 symbols (numeric characters), represented as follows:

2, 2, 3, 0, 2, 2, 2, 0, 2, 2, 1, 4

Correspondingly, encoded data (E2) delivered by the encoder pursuant to embodiments of the present disclosure can be represented as follows:

2, 5, 3, 0, 0, 1, 4, 2, 0, 3, 2, 0

Hereinabove, a first symbol '2' indicates a mode symbol, a second symbol '5' indicates the count of non-mode symbols. The second symbol '5' is followed by a non-mode stream, which is followed by a runs stream.

In the illustrated example, the original non-mode stream cannot contain a value of '2', namely a mode value of '2'. Therefore, optionally, all values greater than '2' are decreased by one to generate a modified non-mode stream. This can be implemented either during encoding or after encoding, before compression.

As a result, a maximum index for the non-mode stream decreases from '4' to '3', and the modified non-mode stream is represented as follows:

2, 0, 0, 1, 3

During subsequent decoding, a given decoder recovers the original non-mode stream from the modified non-mode stream, by increasing all values equal to or greater than the mode value '2' by one. It will be appreciated that this recovering can be implemented before decoding or in connection with decoding.

For the sake of simplicity and clarity, the examples have been provided for a case where there is a single mode symbol in the input data (D1). However, it is to be noted here that embodiments of the present disclosure can be implemented in a similar manner for a case where there are more than one mode symbol in the input data (D1).

In a second aspect, embodiments of the present disclosure provide a method of encoding input data (D1) to generate corresponding encoded data (E2), characterized in that the method includes:

(a) analyzing the input data (D1) to identify at least one mode symbol therein;

(b) generating a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) generating information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) encoding the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

According to an embodiment, the method includes generating symbols in the input data (D1) prior to (a) to (d) by splitting or combining original symbols present in the input data (D1). Optionally, in this regard, the method includes:

(i) splitting at least one or more portions of the input data (D1) to generate one or more new symbols for processing pursuant to (a) to (d); and/or

(ii) combining one or more alphanumeric characters of at least one or more portions of the input data (D1) to generate new combined symbols for processing pursuant to (a) to (d).

According to an embodiment, the method is susceptible to being used recursively. Optionally, in this regard, the method includes recursively processing the first data stream including the non-mode symbols pursuant to (a) to (d).

According to an embodiment, the method includes interleaving data of the first and second data streams to provide a resultant data stream for compression in (d). According to another embodiment, the method includes interleaving data of the first and second data streams as interleaved data blocks to provide a resultant data stream for compression in (d).

According to an embodiment, the method includes delivering the encoded data (E2) to at least one decoder. As an example, the encoded data (E2) is written to a file by a given encoder, and is then read from the file by the at least one decoder. As another example, the encoded data (E2) is streamed from the given encoder to the at least one decoder.

According to an embodiment, the method includes delivering additional information, in the encoded data (E2), relating to delivery of at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, and/or maximum indices of different data.

According to an embodiment, the method includes delivering, to the at least one decoder, at least one index to indicate whether or not the at least one mode symbol is communicated in the encoded data (E2).

According to an embodiment, the method includes encoding the information that is indicative of the at least one mode symbol to generate the encoded data (E2) at least in part as prior information to the at least one decoder.

According to an embodiment, the method includes dynamically selecting, from amongst a plurality of encoding algorithms, an encoding algorithm to be employed in (a) to (d), depending upon one or more properties of the input data (D1), to obtain an improved degree of data compression in the encoded data (E2).

According to an embodiment, the method includes employing range encoding data compression and/or Huffman encoding data compression to compress the second data stream when generating the encoded data (E2).

According to an embodiment, the method includes employing additional encoding methods for encoding one or more portions of the input data (D1), wherein the additional methods include at least one of: Delta encoding, ODelta encoding, RLE, SRLE, ZRLE, Huffman coding, Range coding, EM encoding, Continuum operator encoding. These additional methods are elucidated in greater detail in the foregoing.

In a third aspect, embodiments of the present disclosure provide a decoder for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

(i) the decoder is operable to decode from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) the decoder is operable to insert before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

Optionally, when more than one mode symbol are employed in the encoded data (E2), the decoder is operable to recursively process (i) to (ii) for each mode symbol to generate the decoded data (D3).

According to an embodiment, the decoder is operable to receive additional information, in the encoded data (E2), relating to at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, and/or maximum indices of different data. Optionally, in this regard, the decoder is operable to use the additional information for proper decoding of the encoded data (E2).

Optionally, when the encoded data (E2) has been encoded using additional encoding methods, the decoder is operable to decode the encoded data (E2) in a reverse order of processing steps to an order of processing steps employed during encoding of original input data (D1) to generate the encoded data (E2) by a corresponding encoder. In this regard, optionally, the decoder is operable to employ additional decoding methods, wherein the additional decoding methods include at least one of: Delta decoding, ODelta decoding, RLE decoding, SRLE decoding, ZRLE decoding, Huffman decoding, Range decoding, Entropy Modifying (EM) decoding, Continuum operator decoding. Thus, the decoding methods are inverse methods of the coding methods used in the encoder; however, when transcoding is implemented in the decoder, additional or alternative methods are employed in the decoder, such that the data D1, D3 are then mutually different but often include substantially similar content; for example, the decoder is implemented on a mobile wireless device, and applies to transcoding to reformat video content present in the input data (D1), and encoded into the encoded data (E2), so that the content can be presented in an appropriate manner on a miniature pixel display of the mobile wireless device. These additional, or alternative methods are, for example, elucidated in greater detail in the foregoing.

Furthermore, optionally, the aforementioned decoder is useable with other known decoders; for example, in conjunction with a block decoder as described in reference [7].

In a fourth aspect, embodiments of the present disclosure provide a method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

(i) decoding from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) inserting before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

Optionally, when more than one mode symbol are employed in the encoded data (E2), the method includes recursively processing (i) to (ii) for each mode symbol to generate the decoded data (D3).

According to an embodiment, the method includes receiving additional information, in the encoded data (E2), relating to at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, and/or maximum indices of different data. Optionally, in this regard, the method includes using the additional information for proper decoding of the encoded data (E2).

Optionally, when the encoded data (E2) has been encoded using additional encoding methods, the method includes decoding the encoded data (E2) in a reverse order of processing steps to an order of processing steps employed during encoding of original input data (D1) to generate the encoded data (E2) by a corresponding encoder. In this regard, optionally, the method includes employing additional decoding methods, wherein the additional decoding methods include at least one of: Delta decoding, ODelta decoding, RLE decoding, SRLE decoding, ZRLE decoding, Huffman decoding, Range decoding, Entropy Modifying (EM) decoding, Continuum operator decoding. Such additional decoding methods are described in documents mentioned in the foregoing, and also in the references.

In a fifth aspect, embodiments of the present disclosure provide a codec including at least one encoder for encoding input data (D1) to generate corresponding encoded data (E2) pursuant to embodiments of the present disclosure, and at least one decoder for decoding the encoded data (E2) to generate corresponding decoded data (D3) pursuant to embodiments of the present disclosure.

An example codec has been provided in conjunction with FIG. 1 as explained in more detail below. The codec includes at least one encoder **110** for encoding input data (D1) to generate corresponding encoded data (E2), and at least one decoder **120** for decoding the encoded data (E2) to generate corresponding decoded data (D3). Collectively, the at least one encoder **110** and the at least one decoder **120** constitute a codec **130**.

Optionally, the decoded data (D3) is exactly similar to the input data (D1), as in a lossless mode of operation. Alternatively, optionally, the decoded data (D3) is approximately similar to the input data (D1), as in a lossy mode of operation. Yet alternatively, optionally, the decoded data (D3) is different to the input data (D1), for example by way of a transformation, for example transcoding, but retains substantially similar information present in the input data (D1); for example, the decoded data (D3) is usefully made different to the input data (D1) when reformatting of the decoded data (D3) is also required, for example to be compatible with different types of communication platforms, software layers, communication devices, and so forth.

The at least one encoder includes a data processing arrangement for processing the input data (D1) to generate the corresponding encoded data (E2) pursuant to embodiments of the present disclosure. Optionally, the data processing arrangement of the at least one encoder is implemented by employing at least one Reduced Instruction Set Computing (RISC) processor that is operable to execute program instructions as elucidated earlier.

Furthermore, optionally, the at least one encoder **110** is operable to communicate the encoded data (E2) to a data server and/or data storage for storing in a database. The data server and/or data storage is arranged to be accessible to the at least one decoder **120**, which is beneficially compatible with the at least one encoder **110**, for subsequently decoding the encoded data (E2).

In some examples, the at least one decoder is optionally operable to access the encoded data (E2) from the data server and/or data storage.

In alternative examples, the at least one encoder **110** is optionally operable to stream the encoded data (E2) to the at least one decoder **120**, either via a communication network or via a direct connection. Moreover, it is to be noted that a device equipped with a hardware-based or software-based encoder can also communicate directly with another device equipped with a hardware-based or software-based decoder.

In yet other alternative examples, the at least one decoder **120** is optionally implemented so as to retrieve the encoded data (E2) from a non-transitory (namely non-transient) computer-readable storage medium, such as a hard drive and a Solid-State Drive (SSD).

The at least one decoder **120** includes a data processing arrangement for processing the encoded data (E2) to generate the corresponding decoded data (D3) pursuant to embodiments of the present disclosure. Optionally, the data processing arrangement of the at least one decoder **120** is implemented by employing at least one RISC processor that is operable to execute program instructions as elucidated earlier; such a RISC processor is capable of performing relatively simpler concatenated operations at a very high speed, and is suitable for decoding data provided in a streamed format, for example in real-time.

When embodiments of the present disclosure are implemented in a multicasting manner, there is a plurality of such decoders that are employed.

Optionally, the codec **130** is implemented within a single device. Alternatively, optionally, the codec **130** is effectively implemented between multiple devices. Optionally, the codec **130** is implemented as custom-design digital hardware, for example via use of one or more Application-Specific Integrated Circuits (ASIC's). Alternatively or additionally, optionally, the codec **130** is implemented using computing hardware that is operable to execute program instructions, for example provided to the computing hardware on a non-transient (non-transitory) machine-readable data carrier.

As an example, the at least one encoder **110** and/or the at least one decoder **120** can be beneficially employed in consumer electronics apparatus, wireless communication apparatus and associated systems, digital cameras, smart phones, tablet computers, personal computers, scientific measuring apparatus, communications equipment, videoconferencing equipment, satellites, but not limited thereto.

In a sixth aspect, embodiments of the present disclosure provide a computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute any of the aforementioned methods pursuant to embodiments of the present disclosure.

Optionally, the computer-readable instructions are downloadable from a software application store, for example, from an "*App store*" to the computerized device.

Next, embodiments of the present disclosure will be described with reference to figures.

Referring to FIG. 1, embodiments of the present disclosure concern:

(i) an encoder **110** for encoding input data (D1) to generate corresponding encoded data (E2), and corresponding methods of encoding the input data (D1) to generate the encoded data (E2);
(ii) a decoder **120** for decoding the encoded data (E2) to generate corresponding decoded data (D3), and corresponding methods of decoding the encoded data (E2) to generate the decoded data (D3); and
(iii) a codec **130** including a combination of at least one encoder and at least one decoder, for example a combination of the encoder **110** and the decoder **120**.

FIG. 1 is merely an example, which does not unduly limit the scope of the claims herein. It is to be understood that the specific designation for the codec **130** is provided as an example and is not to be construed as limiting the codec **130** to specific numbers, types, or arrangements of encoders and decoders. A person skilled in the art will recognize many variations, alternatives, and modifications of embodiments of the present disclosure.

Referring now to FIG. 2, there is provided a flow chart depicting steps of a method of encoding input data (D1) to generate corresponding encoded data (E2), in accordance with an embodiment of the present disclosure. The method is depicted as a collection of steps in a logical flow diagram, which represents a sequence of steps that can be implemented in hardware, software, or a combination thereof, for example as aforementioned.

At a step **202**, the input data (D1) is analyzed to identify at least one mode symbol therein.

At a step **204**, a first data stream and a second data stream are generated, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol. The runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1).

At a step **206**, information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol are generated.

At a step **208**, the information that is indicative of the at least one mode symbol generated at the step **206**, the information that is indicative of the count of the non-mode symbols generated at the step **206**, and the first and second data streams generated at the step **204** are encoded to generate the encoded data (E2).

The steps **202** to **208** are only illustrative and other alternatives can also be provided where one or more steps are added, one or more steps are removed, or one or more steps are provided in a different sequence without departing from the scope of the claims herein.

Referring now to FIG. 3, there is provided a flow chart depicting steps of a method of decoding encoded data (E2) to generate corresponding decoded data (D3), in accordance with an embodiment of the present disclosure. The method is depicted as a collection of steps in a logical flow diagram, which represents a sequence of steps that can be implemented in hardware, software, or a combination thereof, for example as aforementioned.

At a step **302**, information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol are decoded from the encoded data (E2).

At a step **304**, corresponding runs of the at least one mode symbol are inserted before or after the non-mode symbols to generate the decoded data (D3).

The steps **302** to **304** are only illustrative and other alternatives can also be provided where one or more steps are added, one or more steps are removed, or one or more steps are provided in a different sequence without departing from the scope of the claims herein.

Modifications to embodiments of the invention described in the foregoing are possible without departing from the scope of the invention as defined by the accompanying claims. Expressions such as “including”, “comprising”, “incorporating”, “consisting of”, “have”, “is” used to describe and claim the present invention are intended to be construed in a non-exclusive manner, namely allowing for items, components or elements not explicitly described also to be present. Reference to the singular is also to be construed to relate to the plural. Numerals included within parentheses in the accompanying claims are intended to assist understanding of the claims and should not be construed in any way to limit subject matter claimed by these claims.

REFERENCES

[1] Run-length encoding - Wikipedia, the free encyclopedia (accessed March 25, 2015); URL: http://en.wikipedia.org/wiki/Run-length_encoding

[2] Data Encoder, Data Decoder and Method (accessed March 25, 2015); URL: http://gurulogic.com/files/US8823560B1.pdf

[3] Proposal for a new zlib strategy: Z_RLE (accessed March 25, 2015); URL: http://optipng.sourceforge.net/pngtech/z_rle.html

[4] ENCODER APPARATUS, DECODER APPARATUS AND METHOD (accessed March 25, 2015); URL: http://gurulogic.com/files/US8933826B2.pdf

[5] Entropy Modifier and Method (accessed March 25, 2015) URL: http://gurulogic.com/files/US8754791B1.pdf

[6] Encoder and Method (accessed March 25, 2015); URL: http://gurulogic.com/files/US8675731B2.pdf

[7] Decoder and Method (accessed March 25, 2015); URL: http://gurulogic.com/files/US2014044191A1.pdf

CLAIMS

We claim:

1. An encoder (110) for encoding input data (D1) to generate corresponding encoded data (E2), characterized in that:

(a) the encoder (110) is operable to analyze the input data (D1) to identify at least one mode symbol therein;

(b) the encoder (110) is operable to generate a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) the encoder (110) is operable to generate information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) the encoder (110) is operable to encode the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

2. The encoder (110) of claim 1, characterized in that the encoder (110) is operable to recursively process the first data stream including the non-mode symbols pursuant to (a) to (d).

3. The encoder (110) of claim 1 or 2, characterized in that, in (d), the encoder (110) is operable to employ range encoding data compression and/or Huffman encoding data compression to compress the second data stream when generating the encoded data (E2).

4. The encoder (110) of claim 1, 2 or 3, characterized in that the encoder (110) is operable to deliver additional information, in the encoded data (E2), relating to delivery of at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, maximum indices of different data.

5. The encoder (110) of claim 1, 2, 3 or 4, characterized in that the encoder (110) is operable to encode the information that is indicative of the at least one mode symbol to generate the encoded data (E2) at least in part as prior information to at least one decoder (120).

6. The encoder (110) of any of claims 1 to 5, characterized in that the encoder (110) is operable to generate symbols in the input data (D1) prior to (a) to (d) by splitting or combining original symbols present in the input data (D1).

7. The encoder (110) of claim 6, characterized in that the encoder (110) is operable:

(i) to split at least one or more portions of the input data (D1) to generate one or more new symbols for processing pursuant to (a) to (d); and/or

(ii) to combine one or more alphanumeric characters of at least one or more portions of the input data (D1) to generate new combined symbols for processing pursuant to (a) to (d).

8. The encoder (110) of any of claims 1 to 7, characterized in that the encoder (110) is operable to employ additional encoding methods for encoding one or more portions of the input data (D1), wherein the additional methods include at least one of: Delta encoding, ODelta encoding, RLE encoding, SRLE encoding, ZRLE encoding, Huffman coding, Range coding, Entropy Modifying (EM) encoding, Continuum operator encoding.

9. The encoder (110) of any of claims 1 to 8, characterized in that the encoder (110) is operable to dynamically select, from amongst a plurality of

encoding algorithms, an encoding algorithm to be employed in (a) to (d), depending upon one or more properties of the input data (D1), to obtain an improved degree of data compression in the encoded data (E2).

10. The encoder (110) of any of claims 1 to 9, characterized in that the encoder (110) is operable to interleave data of the first and second data streams to provide a resultant data stream for compression in (d).

11. The encoder (110) of any of claims 1 to 9, characterized in that the encoder (110) is operable to interleave data of the first and second data streams as interleaved data blocks to provide a resultant data stream for compression in (d).

12. The encoder (110) of any of claims 1 to 11, characterized in that the encoder (110) is operable to deliver, to at least one decoder (120), at least one index to indicate whether or not the at least one mode symbol is communicated in the encoded data (E2).

13. A method of encoding input data (D1) to generate corresponding encoded data (E2), characterized in that the method includes:

(a) analyzing the input data (D1) to identify at least one mode symbol therein;

(b) generating a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) generating information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) encoding the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

14. The method of claim 13, characterized in that the method includes recursively processing the first data stream including the non-mode symbols pursuant to (a) to (d).

15. The method of claim 13 or 14, characterized in that, in (d), the method includes employing range encoding data compression and/or Huffman encoding data compression to compress the second data stream when generating the encoded data (E2).

16. The method of claim 13, 14 or 15, characterized in that the method includes delivering additional information, in the encoded data (E2), relating to delivery of at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, maximum indices of different data.

17. The method of claim 13, 14, 15 or 16, characterized in that the method includes encoding the information that is indicative of the at least one mode symbol to generate the encoded data (E2) at least in part as prior information to at least one decoder (120).

18. The method of any of claims 13 to 17, characterized in that the method includes generating symbols in the input data (D1) prior to (a) to (d) by splitting or combining original symbols present in the input data (D1).

19. The method of claim 18, characterized in that the method includes:

(i) splitting at least one or more portions of the input data (D1) to generate one or more new symbols for processing pursuant to (a) to (d); and/or

(ii) combining one or more alphanumeric characters of at least one or more portions of the input data (D1) to generate new combined symbols for processing pursuant to (a) to (d).

20. The method of any of claims 13 to 19, characterized in that the method includes employing additional encoding methods for encoding one or more portions of the input data (D1), wherein the additional methods include at least one of: Delta encoding, ODelta encoding, RLE encoding, SRLE encoding, ZRLE encoding, Huffman coding, Range coding, Entropy Modifying (EM) encoding, Continuum operator encoding.

21. The method of any of claims 13 to 20, characterized in that the method includes dynamically selecting, from amongst a plurality of encoding algorithms, an encoding algorithm to be employed in (a) to (d), depending upon one or more properties of the input data (D1), to obtain an improved degree of data compression in the encoded data (E2).

22. The method of any of claims 13 to 21, characterized in that the method includes interleaving data of the first and second data streams to provide a resultant data stream for compression in (d).

23. The method of any of claims 13 to 21, characterized in that the method includes interleaving data of the first and second data streams as interleaved data blocks to provide a resultant data stream for compression in (d).

24. The method of any of claims 13 to 23, characterized in that the method includes delivering, to at least one decoder (120), at least one index to indicate whether or not the at least one mode symbol is communicated in the encoded data (E2).

25. A decoder (120) for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

(i) the decoder (120) is operable to decode from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) the decoder (120) is operable to insert before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

26. A method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

(i) decoding from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) inserting before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

27. A codec (130) including at least one encoder (110) of any of claims 1 to 12 for encoding input data (D1) to generate corresponding encoded data (E2), and at least one decoder (120) of claim 25 for decoding the encoded data (E2) to generate corresponding decoded data (D3).

28. A computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored

thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute a method as claimed in any of claims 13 to 24 or claim 26.

Amendments to the claims have been made as follows

Claims

1. An encoder (110) for encoding input data (D1) to generate corresponding encoded data (E2), characterized in that:

(a) the encoder (110) is operable to analyze the input data (D1) to identify at least one mode symbol therein;

(b) the encoder (110) is operable to generate a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) the encoder (110) is operable to generate information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) the encoder (110) is operable to encode the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

2. The encoder (110) of claim 1, characterized in that the encoder (110) is operable to recursively process the first data stream including the non-mode symbols pursuant to (a) to (d).

3. The encoder (110) of claim 1 or 2, characterized in that, in (d), the encoder (110) is operable to employ one or more entropy encoding methods to compress the first and second data streams when generating the encoded data (E2).

4. The encoder (110) of claim 1, 2 or 3, characterized in that the encoder (110) is operable to deliver additional information, in the encoded data (E2), relating to delivery of at least one of: probability and index tables, mode

symbols, counts of mode symbols, indices of selected methods, data values, maximum indices of different data.

5. The encoder (110) of claim 1, 2, 3 or 4, characterized in that the encoder (110) is operable to encode and deliver the information that is indicative of the at least one mode symbol and the count of mode symbols as prior information before a remainder of the encoded data (E2) is delivered to at least one decoder (120).

6. The encoder (110) of any of claims 1 to 5, characterized in that the encoder (110) is operable to generate symbols in the input data (D1) prior to (a) to (d) by splitting or combining original symbols present in the input data (D1).

7. The encoder (110) of claim 6, characterized in that the encoder (110) is operable:

(i) to split at least one or more portions of the input data (D1) to generate one or more new symbols for processing pursuant to (a) to (d); and/or

(ii) to combine one or more alphanumeric characters of at least one or more portions of the input data (D1) to generate new combined symbols for processing pursuant to (a) to (d).

8. The encoder (110) of any of claims 1 to 7, characterized in that the encoder (110) is operable to employ additional encoding methods for encoding one or more portions of the input data (D1), wherein the additional methods include at least one of: Delta encoding, ODelta encoding, RLE encoding, SRLE encoding, ZRLE encoding, Huffman coding, Range coding, Entropy Modifying (EM) encoding, Continuum operator encoding.

9. The encoder (110) of any of claims 1 to 8, characterized in that the encoder (110) is operable to dynamically select, from amongst a plurality of encoding algorithms, an encoding algorithm to be employed in (a) to (d), depending upon one or more properties of the input data (D1), to obtain an improved degree of data compression in the encoded data (E2).

10. The encoder (110) of any of claims 1 to 9, characterized in that the encoder (110) is operable to interleave data of the first and second data streams to provide a resultant data stream for compression in (d).

11. The encoder (110) of any of claims 1 to 9, characterized in that the encoder (110) is operable to interleave data of the first and second data streams as interleaved data blocks to provide a resultant data stream for compression in (d).

12. The encoder (110) of any of claims 1 to 11, characterized in that the encoder (110) is operable to deliver, to at least one decoder (120), at least one index to indicate whether or not the at least one mode symbol is communicated in the encoded data (E2).

13. A method of encoding input data (D1) to generate corresponding encoded data (E2), characterized in that the method includes:

(a) analyzing the input data (D1) to identify at least one mode symbol therein;

(b) generating a first data stream and a second data stream, wherein the first data stream includes non-mode symbols, and the second data stream includes runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within the input data (D1);

(c) generating information that is indicative of a count of the non-mode symbols in the first data stream and information that is indicative of the at least one mode symbol; and

(d) encoding the information that is indicative of the at least one mode symbol, the information that is indicative of the count of the non-mode symbols, the first data stream including the non-mode symbols and the second data stream including the runs of the at least one mode symbol, to generate the encoded data (E2).

14. The method of claim 13, characterized in that the method includes recursively processing the first data stream including the non-mode symbols pursuant to (a) to (d).

15. The method of claim 13 or 14, characterized in that, in (d), the method includes employing one or more entropy encoding methods to compress the first and second data streams when generating the encoded data (E2).

16. The method of claim 13, 14 or 15, characterized in that the method includes delivering additional information, in the encoded data (E2), relating to delivery of at least one of: probability and index tables, mode symbols, counts of mode symbols, indices of selected methods, data values, maximum indices of different data.

17. The method of claim 13, 14, 15 or 16, characterized in that the method includes encoding and delivering the information that is indicative of the at least one mode symbol and the count of mode symbols as prior information before a remainder of the encoded data (E2) is delivered to at least one decoder (120).

18. The method of any of claims 13 to 17, characterized in that the method includes generating symbols in the input data (D1) prior to (a) to (d) by splitting or combining original symbols present in the input data (D1).

19. The method of claim 18, characterized in that the method includes:
(i) splitting at least one or more portions of the input data (D1) to generate one or more new symbols for processing pursuant to (a) to (d); and/or
(ii) combining one or more alphanumeric characters of at least one or more portions of the input data (D1) to generate new combined symbols for processing pursuant to (a) to (d).

20. The method of any of claims 13 to 19, characterized in that the method includes employing additional encoding methods for encoding one or more portions of the input data (D1), wherein the additional methods include at least one of: Delta encoding, ODelta encoding, RLE encoding, SRLE encoding, ZRLE

encoding, Huffman coding, Range coding, Entropy Modifying (EM) encoding, Continuum operator encoding.

21. The method of any of claims 13 to 20, characterized in that the method includes dynamically selecting, from amongst a plurality of encoding algorithms, an encoding algorithm to be employed in (a) to (d), depending upon one or more properties of the input data (D1), to obtain an improved degree of data compression in the encoded data (E2).

22. The method of any of claims 13 to 21, characterized in that the method includes interleaving data of the first and second data streams to provide a resultant data stream for compression in (d).

23. The method of any of claims 13 to 21, characterized in that the method includes interleaving data of the first and second data streams as interleaved data blocks to provide a resultant data stream for compression in (d).

24. The method of any of claims 13 to 23, characterized in that the method includes delivering, to at least one decoder (120), at least one index to indicate whether or not the at least one mode symbol is communicated in the encoded data (E2).

25. A decoder (120) for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

(i) the decoder (120) is operable to decode from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) the decoder (120) is operable to insert before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

26. A method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

(i) decoding from the encoded data (E2) information that is indicative of at least one mode symbol, information that is indicative of a count of non-mode symbols, a first data stream including non-mode symbols and a second data stream including runs of the at least one mode symbol, wherein the runs of the at least one mode symbol are indicative of occurrences of the at least one mode symbol before or after the non-mode symbols within original input data (D1); and

(ii) inserting before or after the non-mode symbols corresponding runs of the at least one mode symbol to generate the decoded data (D3).

27. A codec (130) including at least one encoder (110) of any of claims 1 to 12 for encoding input data (D1) to generate corresponding encoded data (E2), and at least one decoder (120) of claim 25 for decoding the encoded data (E2) to generate corresponding decoded data (D3).

28. A computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute a method as claimed in any of claims 13 to 24 or claim 26.

Intellectual Property Office

Application No:	GB1510127.2	**Examiner:**	Mr Steven Davies
Claims searched:	1-28	**Date of search:**	22 December 2015

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
A	-	CN 1949670 A (Zhongxing Comm) see online abstract
A	-	WO2010/115789 A1 (Thomson Licensing)
A	-	WO2005/043921 A1 (Sightspeed)

Categories:

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^X :

Worldwide search of patent documents classified in the following areas of the IPC

G06F; H03M; H04L; H04N

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC, INSPEC

International Classification:

Subclass	Subgroup	Valid From
H04L	0029/06	01/01/2006
H03M	0007/48	01/01/2006