(12) UK Patent (19) GB (11) 2507603 (13) B

(45) Date of B Publication **01.10.2014**

(54) Title of the Invention: Data encoder, data decoder and method

(51) INT CL: ***H03M 7/46*** (2006.01) ***H04L 29/06*** (2006.01)

(21) Application No: **1303660.3**

(22) Date of Filing: **01.03.2013**

(43) Date of A Publication **07.05.2014**

(56) Documents Cited:
GB 2481870 A **US 20040178933 A1**
"Run-length encoding", retrievable from http://en.wikipedia.org/wiki/Run-length_encoding on or after 24.07.2013
IEEE Signal Processing Letters, pub. IEEE, US, vol. 10, no.3, March 2003, pp.61 - 64, Chengjie Tu et al, "Adaptive runlength coding"

(58) Field of Search:
As for published application 2507603 A viz:
INT CL **H03M, H04L, H04N**
Other: **WPI, EPODOC, INSPEC**
updated as appropriate

(72) Inventor(s):
Ossi Kalevo
Tuomas Mikael Kärkkäinen

(73) Proprietor(s):
Gurulogic Microsystems Oy
Linnankatu 34, Turku 20100, Finland

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

GB 2507603 B

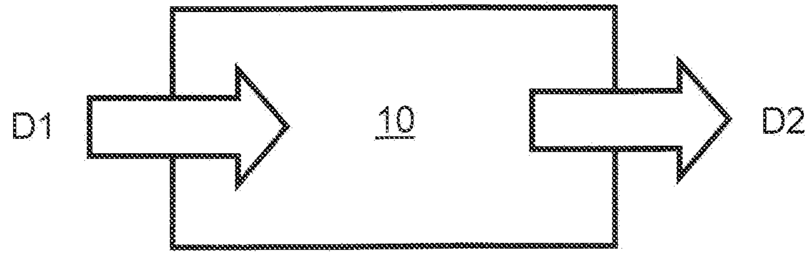

FIG. 1

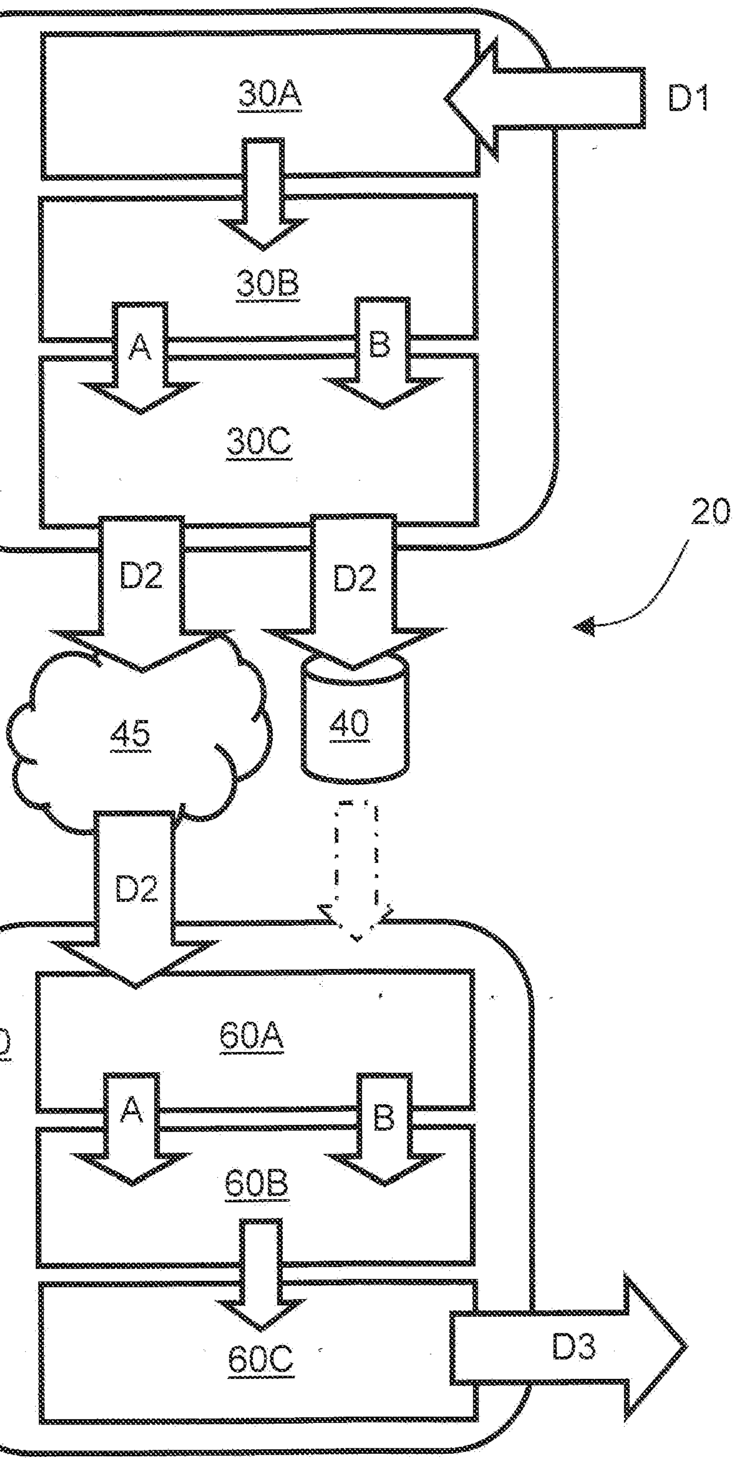

FIG. 2

DATA ENCODER, DATA DECODER AND METHOD

Field of the invention

The present invention relates to data encoders, and corresponding methods of data encoding. Moreover, the present invention relates to data decoders, and corresponding methods of decoding encoded data. Furthermore, the present invention relates to software products recorded on machine-readable data storage media, wherein the software products are executable upon computing hardware for executing aforementioned methods. The data encoder, similarly the data decoder, are useable, for example, as component parts of data communication systems and data supply systems, as well as in electronic consumer products.

Background of the invention

As data is employed to an increasing extent in contemporary society, there is a need for a simple, but effective, method of providing lossless data compression, and corresponding method of providing lossless data de-compression. Conventional known run-length encoding methods of compressing data are only really effective for compressing continuous data, which is generally not so common within data communication systems. Packetized data exchange within data communication systems, for example the Internet, results in transfer of many small packets of data. It is well known to people skilled in the technical art of data compression that run-length encoding (RLE) is capable of providing a better data compression ratio in comparison to other known data compression techniques.

All variations of run-length encoding method employ a counter and a character of bytes which are encoded by such methods. When implementing the methods, the counter is set for each run of such characters. Moreover, there are several variations in the methods regarding how to represent the counter for the character, but these are always encoded into the same output data with the characters of bytes. In consequence, regular known RLE methods have been found to provide a relatively poor data compression ratio.

Literature representing known contemporary art is provided in Table 1.

Table 1: Known technical art

Document ref.	Detail of document
Document D1	"Run-length encoding" – Wikipedia, (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Run-length_encoding
Document D2	"Huffman coding" – Wikipedia, (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Huffman_coding
Document D3	"Run Length Encoding (RLE) – Discussion and Implementation" (accessed 18 December 2012), URL: http://michael.dipperstein.com/rle/index.html
Document D4	"Run Length Encoding" by Arturo Campos (accessed 18 December 2012), URL: http://www.arturocampos.com/ac_rle.html
Document D5	"A Mathematical Theory of Communication" Claude E. Shannon (1948), (accessed 28 November 2012), URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf
Document D6	Shannon's source coding theorem – Wikipedia (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Source_coding_theorem
Document D7	"Entropy" – Wikipedia (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Entropy
Document D8	"Variable-length code" – Wikipedia (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Arithmetic_coding
Document D9	"Arithmetic coding" – Wikipedia (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Arithmetic_coding
Document D10	"Delta encoding" – Wikipedia (accessed 28 November 2012), URL: http://en.wikipedia.org/wiki/Delta_coding

24 01 14

In a published United States patent application US2004/0178933A1 ("*Encoding method and encoding apparatus, and computer program and computer readable storage medium*", Applicant – Canon Kabushiki, inventor – Umeda), there is described an innovation concerning re-encoding encoded data, for example in a format of *PackBits* encoded data, which is expressed by a data format of a runlength code part indicating the runlength of the same data and a data part indicating the data, and a runlength code part indicating the runlength of a different data stream and a data part indicating the different data stream, to the same data format without decoding the encoded data, thereby improving a compression ratio which is achievable. Upon reception of data generated by such a PackBits encoding process, a data division unit separates the data into information indicating the runlength of data, and a data part, and outputs them as num and data. A data processing unit masks a predetermined bit in accordance with an instruction from a monitor unit, and outputs that result to a data combining unit. The data combining unit and a data output unit reconstruct and output data in the PackBits format in accordance with the masked data part and num data.

Moreover, an adaptive runlength encoding innovation is described in a technical publication “Adaptive runlength encoding”, IEEE Signal Processing Letters, pub. IEE, US vol. 10, no. 3, March 2003, pp 61 to 64, Chengjie TU *et al.*

Summary of the invention

The present invention seeks to provide an improved method of compressing input data to generate corresponding compressed output data.

The present invention seeks to provide an improved method of encoding input data to generate corresponding encoded output data.

The present invention also seeks to provide an improved method of data de-compression for receiving compressing input data to generate corresponding de-compressed output data.

The present invention seeks to provide an improved method of decoding input data to generate corresponding decoded output data.

The present invention also seeks to provide encoders, decoders and codecs which are operable to employ aforesaid improved methods.

According to a first aspect of the present invention, there is provided an encoder as claimed in appended claim 1: there is provided an encoder for encoding input data (D1) to generate corresponding encoded output data (D2), wherein the encoder includes a data processing arrangement for generating a run-length encoded (RLE) representation of the input data (D1), characterized in that the encoder is operable to split the run-length encoded (RLE) representation into a plurality of parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters, and the encoder is operable to encode the plurality of parts (A, B) separately to generate the encoded output data (D2).

The invention is of advantage in that the encoder is capable of providing a higher degree of substantially lossless compression in comparison to known types of encoder.

Optionally, the encoder is operable to generate the encoded output data (D2) in a compressed form relative to the input data (D1).

Optionally, the encoder is operable to encode the plurality of parts (A, B) to generate the encoded output data (D2) by employing at least one of: variable length coding (VLC), Huffman coding.

Optionally, the encoder is operable to include one or more markers in the encoded output data (D2) for indicating occurrence of encoded data corresponding to each of the plurality of parts (A, B).

Optionally, the encoder is arranged such that the run-length encoded (RLE) representation is implemented in a substantially lossless manner.

According to a second aspect of the invention, there is provided a method of encoding input data (D1) in an encoder to generate corresponding encoded output data (D2), wherein the method includes using a data processing arrangement to generate a run-length encoded (RLE) representation of the input data (D1), characterized in that the method further includes:

(a) using the encoder to split the run-length encoded (RLE) representation into a plurality of parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters; and

(b) using the encoder to encode the plurality of parts (A, B) separately to generate the encoded output data (D2).

Optionally, the method includes using the encoder to generate the encoded output data (D2) in a compressed form relative to the input data (D1).

Optionally, the method includes using the encoder to encode the plurality of parts (A, B) to generate the encoded output data (D2) by employing at least one of: variable length coding (VLC), Huffman coding.

Optionally, the method includes using the encoder to include one or more markers in the encoded output data (D2) for indicating occurrence of encoded data corresponding to each of the plurality of parts (A, B).

Optionally, the method includes implementing the run-length encoded (RLE) representation in a substantially lossless manner.

According to a third aspect of the invention, there is provided a decoder for decoding encoded input data (D2) to generate corresponding decoded output data (D3), characterized in that the decoder includes a data processing arrangement for decoding the encoded input data (D2) into a plurality of separate parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters, and for combining the plurality of parts (A, B) together to generate combined data for decoding via a run-length decoding process to generate the decoded output data (D3).

Optionally, the decoder is operable to generate the decoded output data (D3) in a decompressed form relative to the encoded input data (D2).

Optionally, the decoder is operable to decode the encoded input data (D2) to generate the plurality of parts (A, B) by employing at least one of: inverse variable length coding (VLC), inverse Huffman coding.

Optionally, the decoder is operable to indentify one or more markers in the encoded input data (D2) for determining occurrence of encoded data corresponding to each of the plurality of parts (A, B).

Optionally, the decoder is arranged such that the run-length decoding process is implemented in a substantially lossless manner.

According to a fourth aspect of the invention, there is provided a method of decoding encoded input data (D2) in a decoder to generate corresponding decoded output data (D3), wherein the method includes using a data processing arrangement, characterized in that the method includes:

(a) using the decoder to decode the encoded input data (D2) into a plurality of separate parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters;

(b) using the decoder to combine the plurality of parts (A, B) to generate corresponding combined data; and

(c) using a run-length decoding process the apply a run-length decoding process to the combined data to generate the decoded output data (D3).

Optionally, the method includes using the decoder to generate the decoded output data (D3) in a decompressed form relative to the encoded input data (D2).

Optionally, the method includes using the decoder to decode the plurality of parts (A, B) to generate the decoded output data (D3) by employing at least one of: inverse variable length coding (VLC), inverse Huffman coding.

Optionally, the method includes using the decoder to identify one or more markers in the encoded input data (D2) for determining occurrence of encoded data corresponding to each of the plurality of parts (A, B).

Optionally, the method includes implementing the run-length decoding process in a substantially lossless manner.

According to a fifth aspect of the invention, there is provided a codec including an encoder pursuant to the first aspect of the invention for encoding input data (D1) to generate corresponding encoded data (D2), and a decoder pursuant to the third aspect of the invention for decoding the encoded output data (D2) to generate corresponding output decoded data (D3).

According to a sixth aspect of the invention, there is provided a software product recorded on machine-readable data storage media, characterized in that the software product is executable upon computing hardware for executing a method pursuant to the second aspect of the invention.

According to a seventh aspect of the invention, there is provided a software product recorded on machine-readable data storage media, characterized in that the software product is executable upon computing hardware for executing a method pursuant to the fourth aspect of the invention.

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

Description of the diagrams

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

FIG. 1 is an illustration of an encoder pursuant to the present invention; and

FIG. 2 is an illustration of a codec pursuant to the present invention.

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. When a number is non-underlined and accompanied by an associated arrow, the non-underlined number is used to identify a general item at which the arrow is pointing.

Description of embodiments of the invention

In overview, the present invention is based upon a run-length encoding (RLE) method, but provides a significant improvement thereupon.

RLE methods are contemporarily employed to encode various types of graphical information, for example multimedia data formats, for example in a lossless manner. However, such RLE methods do not perform well for continuous-tone images, for example photographs, as well as non-formalized data. Nevertheless, convention RLE is employed in apparatus such as facsimile machines, usually in combination with Huffman-coding and employing standard contemporary formats such as BMP, GIF, PCX, TIFF and such like.

Embodiments of the present invention are concerned with a simple solution for achieving improved data compressing and encoding which provides an enhanced data compression ratio in comparison to contemporary known RLE methods. Moreover, embodiments of the present invention employ a counter, akin to RLE, but with a difference that associated characters are written separately to their own respective data output parts, namely part A and part B respectively. Such separation in mutually different parts is in contradistinction to contemporary RLE approaches where character definition and run-length information are intermixed together.

Thus, embodiments of the present invention concern a novel split run-length method, namely “SRLE” which is effective for both pre-processing and post-processing of continuous data, namely for all types of 1D, 2D and 3D data, for example including graphics, images, video, audio, text, and binary data. The split run-length method is updatable for networking purposes, for implementation in devices, microprocessors and similar. Moreover, the split run-length method is susceptible to being implemented using one or more software products stored on machine-readable data storage media, wherein the software products are executable upon computing hardware.

Referring to FIG. 1, there is provided an illustration of an encoder **10**, for example implemented by way of digital hardware and/or one or more software products stored on machine-readable data-storage media and executable upon computing hardware, which is operable to receive input data D1 and generate corresponding encoded output data D2. For such an encoder **10**, it is possible to calculate an entropy E1 of the input data D1, as well as an entropy E2 of the output data D2. The entropies E1, E2 are manifest in a number of bits required to provide a given set of data.

For elucidating a manner of operation of the encoder **10**, an example run length (RL) is usually expressed in one or more byte values. However, for elucidating embodiments of the present invention, alpha-numerical characters will be described here. The input data D1 is represented by a string of characters as follows:

WWW WWW WWW WWW BWW WWW WWW WWW WBB BWW WWW WWW WWW WWW WWW WWW WWW WBW WWW WWW WWW WWW W

Eq. 1

The entropy E1 associated with such a string of characters is 62 “W” characters and 5 “B” characters which is susceptible to being represented by 25.66 bits, as calculable from Equation 2 (Eq. 2) and Equation 3 (Eq. 3):

$$L = \frac{E}{\log_{10} N} \qquad \text{Eq. 2}$$

Shannon’s theory has indentified that an average length *L* of a shortest possible representation to encode a given message in a given alphabet is their entropy *E* divided by a logarithm of a number of symbols *N* present in the alphabet.

$$M_B = \frac{E}{\log_{10}(2)} = 25.66 \quad \text{bits} \qquad \text{Eq. 3}$$

The series of characters above is susceptible to being encoding using two alternative known RLE methods to generate the output data D2 as an RLE series:

12W 1B 12W 3B 24W 1B 14W Eq. 4

with an associated calculated entropy E2 corresponding to 26.55 bits, alternatively the output data D2 as an RLE series:

WW10BWW10BB1WW22BWW12 Eq. 5

with an associated calculated entropy E2 corresponding to 35.49 bits.

Equation 4 (Eq. 4) is a simple expression of a given character preceded by a number indicative of occurrences of the given character, namely 12 “W” followed by 1 “B” followed by 12 “W” and so forth. Conversely, in Equation 5 (Eq. 5), the characters represent corresponding characters in sequence in Equation 1 (Eq. 1) followed an indication of occurrences thereof thereafter.

Such known RLE methods result in a greater entropy in the encoded output data D2 than present in the input data D1, namely E2 > E1, although there occurs a reduction in a total number of symbols present in the output data D2. It is known to compress RLE data by applying further coding thereto, for example Huffman coding, variable length coding (VLC), but result in sub-optimal data compression results. Such sub-optimal performance arises on account of a mismatch of formats.

It will be appreciated that Equations 4 (Eq. 4) and Equation 5 (Eq. 5) are a mixture of characters, denoted by part A, and a counter associated with the characters, namely part B; such parts are designated in Table 2 in respect of Equations 1 (Eq. 1), Equation 4 (Eq. 4) and Equation 5 (Eq. 5):

<u>Table 2:</u> Parts A and B pursuant to the present invention

Eq. 1	WWW WWW WWW WWW	B	WW WWW WWW WWW W	BB B	WW WWW WWW WWW WWW WWW WWW WWW W	B	W WWW WWW WWW WWW W
RLE method 1: part B (Eq. 4)	12	1	12	3	24	1	14
RLE method 1: part A (Eq. 4)	W	B	W	B	W	B	W
RLE method 2:Part B (Eq.	10		10	1	22		12

5)							
RLE method 2:Part A (Eq. 5)	WW	B	WW	BB	WW	B	WW

When the encoder **10** functions pursuant to the present invention, the output data D2 is not in a mixed form as in Equations 4 and 5 (Eq. 4 and Eq. 5), but part A and part B are expressed separately, namely are split, and their mutual correspondence is defined by a structure of the encoded data D2, for example defined by one or more markers.

Pursuant to an example embodiment of the present invention, in Equation 1 (Eq. 1), the input data D1 has associated therewith the Entropy E1 of 25.66 bits. Equation 1 sequence of characters is run-length encoded (RLE) into two put parts:

Part A: 12 1 12 3 24 1 14

Part B: W B W B W B W,

wherein Part A has an associated calculated entropy E_A of 15.65 bits, and Part B has an associated calculated entropy E_B of 6.90 bits. Such parts are derived from Equation 4 (Eq. 4) in the foregoing. The total sum of the entropies E_A, E_B is 22.55 bits, which is 3.11 bits less, namely 12% less, than pertains to the input data D1, and 14.00 bits less, namely 38% less, than code generated by the known conventional RLE, namely 36.55 bits. Thus, beneficial data compression is susceptible to being achieved pursuant to the present invention by splitting counter and character information generated by RLE, and then applying a compression method, for example VLC coding, Huffman coding or similar.

Pursuant to another example embodiment of the present invention, in Equation 1 (Eq. 1), the input data D1 has associated therewith the Entropy E1 of 25.66 bits. Equation 1 sequence of characters is run-length encoded (RLE) into two put parts:

Part A: 10 10 1 22 12

Part B: W W B W W B B W W B W W,

wherein Part A has an associated calculated entropy E_A of 9.61 bits, and Part B has an associated calculated entropy E_B of 11.02 bits. Such parts are derived from

Equation 5 (Eq. 5) in the foregoing. The total sum of the entropies E_A, E_B is 20.63 bits, which is 5.03 bits less, namely 20% less, than pertains to the input data D1, and 14.86 bits less, namely 42% less, than code generated by the known conventional RLE, namely 36.55 bits. Thus, beneficial data compression is susceptible to being achieved pursuant to the present invention by splitting counter and character information generated by RLE, and then applying a compression method, for example VLC coding, Huffman coding or similar.

The aforementioned example embodiments clearly illustrate an advantage derived from applying a RLE method in the input data D1 to generate corresponding RLE data, then splitting such RLE data into a character part, namely aforesaid part B, and a corresponding run-length count part, namely aforesaid part A, wherein data pertaining to parts A and B are mutually split apart, and then applying encoding to the parts separately to generate the encoded output data D2. Optionally, additional codes, for example one or more markers, are included to indicate which portions of the encoded output data D2 correspond to part A and which correspond to part B, namely for assisting corresponding subsequent decoding activities. Such data encoding processing is beneficially implemented in the encoder **10** of FIG. 1 implemented functionally in a manner as depicted in FIG. 2.

In FIG. 2, there is shown a codec indicated generally by **20**. The codec **20** includes the encoder **10** arranged to implement a method of encoding the input data D1 pursuant to the present invention to generate corresponding encoded output data D2. The encoder **10** is beneficially implemented in digital hardware, for example as computing hardware which is operable to execute one or more software products recorded on machine-readable data storage media. The encoder **10** is operable to perform three functions as follows:

(a) a first stage **30A** of the encoder **10** is operable to process the input data D1 to determine its characters, or generate a character representation thereof, and to generate corresponding run-length counts;

(b) a second stage **30B** of the encoder **10** is operable to split the characters and their corresponding run-length counts generated in the first stage **30A** into two parts, namely parts B and A respectively, in a structured manner; and

(c) a third stage **30C** of the encoder **10** is operable to encode data of the parts A and B, for example via Huffman or variable length coding (VLC), to generate the output encoded data D2.

The output data D2 is optionally stored on a data carrier **40**, for example an optical data storage disc memory. Alternatively, the output data D2 is communicated via a data communication network **45**, for example the Internet, to a decoder **50** of the codec **20**. The decoder **50** is beneficially implemented in digital hardware, for example as computing hardware which is operable to execute one or more software products recorded on machine-readable data storage media. Moreover, the decoder **50** is operable to perform an inverse of processing operations executed by the encoder **10** when in operation, namely the decoder **50** is operable to perform three functions as follows:

(i) a first stage **60A** of the decoder **50** is operable to receive the encoded data D2 and to decode data of the parts A and B, for example via inverse Huffman or inverse variable length coding (VLC), to regenerate the parts A and B;

(ii) a second stage **60B** of the decoder **50** is operable to combine the regenerated parts A and B from the first stage **60A** to generate the characters and their corresponding run-length counts as mixed data; and

(iii) a third stage **60C** of the decoder **50** is operable to apply inverse run-length encoding to the mixed data to regenerate therefrom decoded output data D3.

Optionally, at least one of the encoder **10** and the decoder **50** is implemented in electronics consumer products, for example televisions, smart phones, fax machines, tablet computers, personal computers (PC's), video recorders, video players, digital cameras, video-conferencing systems, surveillance equipment, portable personal multimedia players to mention merely a few practical applications.

Although splitting of encoded data into two parts, namely parts A and B, is described in the foregoing in respect of the present invention, it will be appreciated that splitting into more than two parts is optionally possible for multi-dimensional data. Moreover, although combining of decoded data corresponding to two parts, namely parts A and B, is described in the foregoing in respect of the present invention, it will be appreciated that combining of more than two parts is optionally possible for multi-dimensional data.

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.

CLAIMS

1. An encoder (10) for encoding input data (D1) to generate corresponding encoded output data (D2), wherein the encoder (10) includes a data processing arrangement for generating a run-length encoded (RLE) representation of the input data (D1), characterized in that the encoder (10) is operable to split the run-length encoded (RLE) representation into a plurality of parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters, and the encoder (10) is operable to encode the plurality of parts (A, B) separately to generate the encoded output data (D2).

2. An encoder (10) as claimed in claim 1, characterized in that the encoder (10) is operable to generate the encoded output data (D2) in a compressed form relative to the input data (D1).

3. An encoder (10) as claimed in claim 1 or 2, characterized in that the encoder (10) is operable to encode the plurality of parts (A, B) to generate the encoded output data (D2) by employing at least one of: variable length coding (VLC), Huffman coding.

4. An encoder (10) as claimed in any one of the preceding claims, characterized in that the encoder (10) is operable to include one or more markers in the encoded output data (D2) for indicating occurrence of encoded data corresponding to each of the plurality of parts (A, B).

5. An encoder (10) as claimed in any one of the preceding claims, characterized in that the run-length encoded (RLE) representation is implemented in a substantially lossless manner.

6. A method of encoding input data (D1) in an encoder (10) to generate corresponding encoded output data (D2), wherein the method includes using a data processing arrangement to generate a run-length encoded (RLE) representation of the input data (D1), characterized in that the method further includes:

(a) using the encoder (10) to split the run-length encoded (RLE) representation into a plurality of parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters; and

(b) using the encoder (10) to encode the plurality of parts (A, B) separately to generate the encoded output data (D2).

7. A method as claimed in claim 6, characterized in that the method includes using the encoder (10) to generate the encoded output data (D2) in a compressed form relative to the input data (D1).

8. A method as claimed in claim 6 or 7, characterized in that the method includes using the encoder (10) to encode the plurality of parts (A, B) to generate the encoded output data (D2) by employing at least one of: variable length coding (VLC), Huffman coding.

9. A method as claimed in any one of claims 6 to 8, characterized in that the method includes using the encoder (10) to include one or more markers in the encoded output data (D2) for indicating occurrence of encoded data corresponding to each of the plurality of parts (A, B).

10. A method as claimed in any one of claims 6 to 9, characterized in that the method includes implementing the run-length encoded (RLE) representation in a substantially lossless manner.

11. A decoder (50) for decoding encoded input data (D2) to generate corresponding decoded output data (D3), characterized in that the decoder (50) includes a data processing arrangement for decoding the encoded input data (D2) into a plurality of separate parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters, and for combining the plurality of parts (A, B) together to generate combined data for decoding via a run-length decoding process to generate the decoded output data (D3).

12. A decoder (50) as claimed in claim 11, characterized in that the decoder (50) is operable to generate the decoded output data (D3) in a decompressed form relative to the encoded input data (D2).

13. A decoder (50) as claimed in claim 11 or 12, characterized in that the decoder (50) is operable to decode the encoded input data (D2) to generate the plurality of parts (A, B) by employing at least one of: inverse variable length coding (VLC), inverse Huffman coding.

14. A decoder (50) as claimed in claim 11, 12 or 13, characterized in that the decoder (50) is operable to indentify one or more markers in the encoded input data (D2) for determining occurrence of encoded data corresponding to each of the plurality of parts (A, B).

15. A decoder (50) as claimed in any one of claims 11 to 14, characterized in that the run-length decoding process is implemented in a substantially lossless manner.

16. A method of decoding encoded input data (D2) in a decoder (50) to generate corresponding decoded output data (D3), wherein the method includes using a data processing arrangement, characterized in that the method includes:

(a) using the decoder (50) to decode the encoded input data (D2) into a plurality of separate parts (A, B), wherein at least one part is associated with characters and at least another part is associated counters representative of occurrence of the characters;

(b) using the decoder (50) to combine the plurality of parts (A, B) to generate corresponding combined data; and

(c) using a run-length decoding process the apply a run-length decoding process to the combined data to generate the decoded output data (D3).

17. A method as claimed in claim 16, characterized in that the method includes using the decoder (50) to generate the decoded output data (D3) in a decompressed form relative to the encoded input data (D2).

18. A method as claimed in claim 16 or 17, characterized in that the method includes using the decoder (50) to decode the plurality of parts (A, B) to generate the decoded output data (D3) by employing at least one of: inverse variable length coding (VLC), inverse Huffman coding.

19. A method as claimed in any one of claims 16 to 18, characterized in that the method includes using the decoder (50) to identify one or more markers in the encoded input data (D2) for determining occurrence of encoded data corresponding to each of the plurality of parts (A, B).

20. A method as claimed in any one of claims 16 to 19, characterized in that the method includes implementing the run-length decoding process in a substantially lossless manner.

21. A codec (20) including an encoder (10) as claimed in any one of claims 1 to 5 for encoding input data (D1) to generate corresponding encoded data (D2), and a decoder (50) as claimed in any one or claims 11 to 15 for decoding the encoded output data (D2) to generate corresponding output decoded data (D3).

22. A software product recorded on machine-readable data storage media, characterized in that the software product is executable upon computing hardware for executing a method as claimed in any one of claims 6 to 10.

23. A software product recorded on machine-readable data storage media, characterized in that the software product is executable upon computing hardware for executing a method as claimed in any one of claims 16 to 20.