

(19) **DANMARK**



Patent- og
Varemærkestyrelsen

(12)

Oversættelse af europæisk patentskrift

(10) **DK/EP 2933756 T3**

-
- (51) Int.Cl.: **G 06 K 9/46 (2006.01)** **H 03 M 7/40 (2006.01)**
- (45) Oversættelsen bekendtgjort den: **2020-06-02**
- (80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2020-04-29**
- (86) Europæisk ansøgning nr.: **15169825.5**
- (86) Europæisk indleveringsdag: **2003-04-23**
- (87) Den europæiske ansøgnings publiceringsdag: **2015-10-21**
- (30) Prioritet: **2002-04-23 US 374770 P**
- (62) Stamansøgningsnr: **03718500.6**
- (84) Designerede stater: **AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR**
- (73) Patenthaver: **NTT DOCOMO, INC., 11-1, Nagatacho 2-chome, Chiyoda-ku, Tokyo, 100-6150, Japan**
- (72) Opfinder: **Bossen, Frank Jan, 1700 North First Street, Apt. 242, San Jose, California 95112, USA**
- (74) Fuldmægtig i Danmark: **Plougmann Vingtoft A/S, Strandvejen 70, 2900 Hellerup, Danmark**
- (54) Benævnelse: **SYSTEM OG FREMGANGSMÅDE TIL ARITMETISK KODNING OG AFKODNING**
- (56) Fremdragne publikationer:
WO-A1-98/27734
GB-A- 2 330 473
US-B1- 6 195 465
US-B1- 6 198 848
"LINE TRANSMISSION OF NON-TELEPHONE SIGNALS VIDEO CODING FOR LOW BITRATE COMMUNICATION", ITU-T RECOMMENDATION H.263, XX, XX, 2 May 1996 (1996-05-02), pages I-IV,01, XP001025117,
BOSSSEN: "Reducing dec. complexity by an order of magnitude", 3. JVT MEETING; 60. MPEG MEETING; 06-05-2002 - 10-05-2002; FAIRFAX,US; (JOINT VIDEO TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16),, no. JVT-C038, 10 May 2002 (2002-05-10), XP030005146, ISSN: 0000-0442
SALOMON D ED - SALOMON DAVID: "DATA COMPRESSION: THE COMPLETE REFERENCE; Section 2.14, 2.15 : (Adaptive) Arithmetic Coding", 1 January 1998 (1998-01-01), DATA COMPRESSION : THE COMPLETE REFERENCE, SPRINGER, NEW YORK, PAGE(S) 69 - 84, XP002270343, ISBN: 978-0-387-98280-9
ALISTAIR MOFFAT ET AL: "Arithmetic coding revisited", ACM TRANSACTIONS ON INFORMATION SYSTEMS, ASSOCIATION FOR COMPUTING MACHINERY, 2 PENN PLAZA, SUITE 701 NEW YORK NY 10121-0701 USA, vol. 16, no. 3, 1 July 1998 (1998-07-01), pages 256-294, XP058394722, ISSN: 1046-8188, DOI: 10.1145/290159.290162

DESCRIPTION

BACKGROUND OF THE INVENTION

[0001] This application relates to the processing of data, and more particularly, to the encoding and decoding of data sequences in a reduced complexity bound manner.

[0002] It is common to compress a sequence of data (i.e. binary or M-ary information) for sending from a first location to a second location. For example, video data may be generated at the first location, where the data represents a video conference call. The data is typically converted to a sequence of events (i.e. binary events). To deliver the sequence of events to the second location, it is desirable to compress, or encode, the sequence of events to a sequence of information, to reduce the time necessary for transmission of the event sequence to the second location. An arithmetic encoder is typically utilized to compress the sequence of events (i.e. the binary events) to an information sequence of information pieces (i.e. binary bits of information) for sending to the second location. The greater the compression, or encoding efficiency, of the information sequence, the greater the transmission rate of the information sequence to the second location.

[0003] Compression is achieved by generating less than one information piece per event being encoded, where ratios of events to information pieces being encoded may reach 64:1 or even 128:1, depending on the probability distribution of the events. An arithmetic decoder is used at the second location for decompressing the information sequence to the event sequence, thus allowing the video data representing the video conference call to be presented to an individual at the second location.

[0004] Data compression is used for other purposes as well. Another such use may be to increase video quality given an amount of storage media for storing the event sequence. Encoders may be utilized in this example to store video (i.e. representing a movie) on a Compact Disc Read Only Memory (CD-ROM) or Digital Video Disc Read Only Memory (DVD-ROM), or any other storage media including optical, magnetic or otherwise, for replay at a later time. As discussed above, the ratio of events to information pieces encoded may reach 64:1 or 128:1 depending on the probability distribution of the events. At the later time, a decoder is used to decompress the information sequence to the event sequence, thereby allowing the corresponding video or other data to be used.

[0005] However, as encoding efficiency increases at the encoder, a greater amount of processing (i.e. decoding operations) are required at the decoder in order to decode the sequence of information pieces of a given length to the event sequence. Such increased processing requirement may be detrimental, especially where it is desired that the event sequence be provided in a real-time fashion. For example, in the case of an event-sequence representing video, real-time indicates that for the information sequence representing a time

period (i.e. 1 second) of video, that the decoder is capable of receiving the information sequence, and decoding the information sequence to the corresponding event sequence within the time period.

[0006] Sometimes, such real-time processing is not possible with the increased efficiency schemes utilized in conventional encoders.

[0007] This invention is directed to solving one or more of the problems discussed above in a novel and simple manner.

[0008] US 6 198 848 B1 discloses a method of compressing, wherein the length of encoded information for a block is controlled to be the same for each block.

[0009] US 6 195 465 B1 discloses that achieving a fixed rate of compression can support random access to the image region.

[0010] GB 2 330 473 A discloses encoding at fixed length and fixed rate.

BRIEF SUMMARY OF THE INVENTION

[0011] In accordance with one aspect of the invention, a method for arithmetic decoding is provided, the method including the features of claim 1.

[0012] In accordance with another aspect of the invention, a method of arithmetic encoding is provided, the method including the features of claim 2.

[0013] Informative examples for better understanding the present invention are set out below and pertain to aspects of the disclosure.

[0014] In an embodiment, an arithmetic encoder for converting an event sequence including a plurality of events to an information sequence including at least one information piece may be provided. The arithmetic encoder may include: at least one input port for receiving events of the event sequence; at least one output port; a sequencer coupled with the at least one input port and the at least one output port and including a context generator for generating context information for at least one of the received events; a probability estimator coupled with the sequence, for receiving an event of the event sequence and corresponding context information from the sequencer, and generating a probability estimate that the event has a particular value; and a core engine coupled with the sequencer and the probability estimator, for receiving the event of the event sequence from the sequencer, and the probability estimate from the probability estimator, and generating zero or more pieces of the information sequence responsive to the received event and the probability estimate by bounding the ratio of events to information pieces, and providing the zero or more information pieces of the information sequence to the sequencer; wherein the generated information pieces are provided as the

information sequence at the at least one output port.

[0015] In an implementation of the embodiment, the events of the event sequence may be binary events, and the information pieces of the information sequence may be binary bits of information.

[0016] In an implementation of the embodiment, the core processor bounding the ratio of events to information pieces may include bounding the average ratio of events to information pieces.

[0017] In an implementation of the embodiment, the bounded average of events to information pieces may be 4 events per information piece.

[0018] In an implementation of the embodiment, the core processor may include a counter register for tracking a number of events for each information piece, used in bounding the average ratio of events to information pieces to a predetermined ratio value.

[0019] In an implementation of the embodiment, the core processor may generate at least one stuffing information piece within the information sequence where the ratio of events to information pieces exceed the predetermined ratio value.

[0020] In an implementation of the embodiment, the core engine may increment the counter by a predetermined incrementation value for each processed event.

[0021] In an implementation of the embodiment, the predetermined incrementation value may be one.

[0022] In an implementation of the embodiment, the core engine may bound the average ratio of events using the counter register by subtracting the predetermined ratio value from the counter register when an information piece is generated.

[0023] In an implementation of the embodiment, the events may be M-ary in nature.

[0024] In an implementation of the embodiment, the core engine may include a first counter register for tracking a number of events and a second counter register for tracking a number of information pieces, used in bounding the ratio of events to information pieces to a predetermined ratio value.

[0025] In an implementation of the embodiment, the core engine may track the ratio by comparing a value of the first counter with a value of the second counter.

[0026] In an embodiment, a core engine for converting an event sequence including a plurality of events to an information sequence including at least one information piece, may be provided. The core engine may include: a controller for receiving at least one event of the

event sequence and a probability estimate for the at least one event, and for generating zero or more pieces of the information sequence responsive to the received event and the probability estimate by bounding the ratio of events to information pieces.

[0027] In an implementation of the embodiment, the core engine may further include: a counter register coupled with the controller for tracking a number of events for each information piece, used in bounding the ratio of events to information pieces.

[0028] In an implementation of the embodiment, the controller increments the counter by a predetermined incrementation value when an event may be processed.

[0029] In an implementation of the embodiment, the predetermined incrementation value may be one.

[0030] In an implementation of the embodiment, the core engine may bound the ratio of events using the counter register by subtracting the predetermined ratio value from the counter register when an information piece is generated.

[0031] In an implementation of the embodiment, the events of the event sequence may be binary events, and the information pieces of the information may be binary bits of information.

[0032] In an implementation of the embodiment, the controller bounding the ratio of events to information pieces may include bounding the average ratio of events to information pieces.

[0033] In an implementation of the embodiment, the bounded average of events to information pieces may be 4 events per information piece.

[0034] In an implementation of the embodiment, the core processor may generate at least one stuffing information pieces within the information sequence where the average ratio of events to information bits exceeds the predetermined ratio value.

[0035] In an implementation of the embodiment, the core engine may further include: a first counter register coupled with the controller for tracking a number of events; and a second counter coupled with the controller for tracking a number of information pieces; wherein the controller may use the first counter and the second counter in bounding the ratio of events to information pieces.

[0036] In an implementation of the embodiment, the controller bounding the ratio may include comparing a value of the first counter with a value of the second counter.

[0037] In an embodiment, a method of arithmetic encoding to convert an event sequence including a plurality of events to an information sequence including at least one information piece, may be provided. The method of arithmetic encoding may include: receiving at least one event of the event sequence; generating context information for the at least one event;

generating a probability estimate that the event has a particular value responsive to the context information; and generating zero or more information pieces of the information sequence responsive to the received event and the probability estimate by bounding the ratio of events to information pieces.

[0038] In an implementation of the embodiment, the receiving events of the event sequence may include receiving binary events, and the generating zero or more information pieces of the information sequence may include generating zero or more binary bits of information.

[0039] In an implementation of the embodiment, the bounding the ratio of events to information pieces may include bounding the average ratio of events to information pieces.

[0040] In an implementation of the embodiment, the bounding the ratio of events to information pieces may include tracking a number of events for each generated information piece.

[0041] In an implementation of the embodiment, the tracking the number of events for each generated information piece may include tracking the number of events for each generated information piece using a counter.

[0042] In an implementation of the embodiment, the tracking the number of events for each generated information piece using a counter may include tracking the number of events by incrementing the counter by a predetermined incrementation value when an event is processed.

[0043] In an implementation of the embodiment, the incrementing the counter by a predetermined incrementation value may include incrementing the counter by one.

[0044] In an implementation of the embodiment, the tracking the number of events for each generated information piece may include subtracting the predetermined ratio value from the counter register when a piece of information is generated.

[0045] In an implementation of the embodiment, the bounding the average ratio of events to information pieces may include bounding the average ratio of events to information pieces to a value of four.

[0046] In an implementation of the embodiment, the bounding the average ratio of events to information pieces may further include generating at least one stuffing information piece within the information sequence when the ratio of events to information pieces exceeds the predetermined ratio value.

[0047] In an implementation of the embodiment, the tracking the number of events for each generated information piece may include tracking a number of events using a first counter and a number of information pieces using a second counter.

[0048] In an implementation of the embodiment, the tracking the number of events for each generated information piece further may include comparing a value of the first counter with a value of the second counter.

[0049] In an embodiment, a method of arithmetic encoding using an arithmetic encoder, to convert an event sequence including a plurality of events to an information sequence including at least one information piece, may be provided. The method of arithmetic encoding may include: receiving at least one event of the event sequence and a corresponding probability estimate; and generating zero or more information pieces responsive to the event and the corresponding probability estimate, including computing a sub-range value for a particular event value using the corresponding probability estimate, adjusting at least one of a range register and a low register responsive to the received event and the computed sub-range value, and renormalizing a portion of the arithmetic encoder, including bounding an average ratio of events to information pieces, by incrementing a counter register for tracking the average ratio by a predetermined incrementation value where an event is processed, and decrementing the counter register by a predetermined decrementation value where an information piece is generated.

[0050] In an implementation of the embodiment, the decrementing the counter register may include determining whether a value of the counter register is $< \text{zero}$, and decrementing the counter register by the predetermined decrementation value when the value of the counter register is not $< \text{zero}$, and doubling the value of the range register and decrementing the counter register by the predetermined decrementation value when the value of the counter register is $< \text{zero}$.

[0051] In an implementation of the embodiment, the receiving the at least one event may include receiving at least one binary event, and the generating zero or more information pieces includes generating zero or more binary bits.

[0052] In an embodiment, an arithmetic decoder for converting an information sequence including at least one information piece to an event sequence including a plurality of events, may be provided. The arithmetic decoder may include: at least one input port for receiving information pieces of the information sequence; at least one output port; a sequencer coupled with the at least one input port and the at least one output port and including a context generator for generating context information for at least one generated event; a probability estimator coupled with the sequencer, for receiving a generated event of the event sequence and corresponding context information, and determining a probability estimate; and a core engine coupled with the sequencer and the probability estimator, for processing at least one information piece of the information sequence from the sequencer responsive to the probability estimate to generate at least one event by accounting for a bounded ratio of events to information pieces; wherein the generated event is provided to the probability estimator with corresponding context information from the sequencer for updating the probability estimate, and the generated event is provided to the sequencer, for transmittal as the event sequence

from the at least one output port.

[0053] In an implementation of the embodiment, the at least one information piece of the information sequence may be a binary bit, and the events of the event sequence may be binary events.

[0054] In an implementation of the embodiment, the core engine generating at least one event by accounting for a bounded ratio of events to information pieces may include the core engine generating the at least one event by accounting for an average bounded ratio of events to information pieces.

[0055] In an implementation of the embodiment, the core engine accounting for an average bounded ratio of events to information pieces may include the core engine accounting for an average bounded ratio with a value of four events to information pieces.

[0056] In an implementation of the embodiment, the core processor may include a counter register for tracking the number of events to information pieces, used in accounting for the average bounded ratio of events to information pieces being a predetermined average ratio value.

[0057] In an implementation of the embodiment, the core engine may use the counter register for identifying stuffing information pieces in the information sequence.

[0058] In an implementation of the embodiment, the core engine may further include a range register, and the stuffing information pieces in the information sequence may be identified only where the range register is resealed.

[0059] In an implementation of the embodiment, the core engine may increment the counter by a predetermined incrementation value when an event is processed.

[0060] In an implementation of the embodiment, the predetermined incrementation value may be one.

[0061] In an implementation of the embodiment, the core engine may account for the average bounded ratio of events to information pieces using the counter register by subtracting the predetermined ratio value from the counter register when an information piece is processed.

[0062] In an implementation of the embodiment, the generated events of the event sequence may be M-ary in nature.

[0063] In an implementation of the embodiment, the core engine may include a first counter register for tracking a number of information pieces, and a second counter register for tracking a number of events, wherein the first and second counters may be used in accounting for a bounded ratio of events to information pieces.

[0064] In an implementation of the embodiment, the core engine may account for the bounded ratio by comparing a value of the first counter with a value of the second counter.

[0065] In an embodiment, a core engine for converting an information sequence including at least one information piece to an event sequence including a plurality of events may be provided. The core engine may include: a controller for receiving at least one information piece of the information sequence and a probability estimate, and for generating zero or more events of the event sequence responsive to the at least one information piece and probability sequence by accounting for a bounded ratio of events to information pieces.

[0066] In an implementation of the embodiment, the core engine may further include: a counter register coupled with the controller for tracking a number of events for each information piece, used in accounting for the bounded ratio of events to information pieces.

[0067] In an implementation of the embodiment, the controller may increment the counter by a predetermined incrementation value when an event is processed.

[0068] In an implementation of the embodiment, the predetermined incrementation value may be one.

[0069] In an implementation of the embodiment, the core engine may account for the average bounded ratio of events using the counter register by subtracting the predetermined ratio value from the counter register when an information piece is processed.

[0070] In an implementation of the embodiment, the events of the event sequence may be binary events, and the information pieces of the information may be binary bits of information.

[0071] In an implementation of the embodiment, the controller accounting for the bounded ratio of events to information pieces may include accounting for the average bounded ratio of events to information pieces.

[0072] In an implementation of the embodiment, the bounded average of events to information pieces may be 4 events per information piece.

[0073] In an implementation of the embodiment, the core processor may identify zero or more stuffing information pieces in the information sequence using the average bounded ratio of events to information bits.

[0074] In an implementation of the embodiment, the core engine may further include: a first counter for tracking a number of information pieces, a second counter for tracking a number of events, wherein the first and second counters may be used in accounting for the bounded ratio of events to information pieces.

[0075] In an implementation of the embodiment, the core engine may account for the bounded ratio by comparing a value of the first counter with a value of the second counter.

[0076] In an embodiment, a method of arithmetic decoding to convert an information sequence including at least one information piece to an event sequence including a plurality of event sequences, may be provided. The method of arithmetic decoding may include: receiving at least one information piece of the information sequence; generating context information for the at least one event; generating a probability estimate responsive to the context information; and generating zero or more events of the event sequence responsive to the at least one received information piece and the probability estimate by accounting for a bounded ratio of events to information pieces.

[0077] In an implementation of the embodiment, the receiving at least one information piece may include receiving at least one binary bit of information, and the generating zero or more events may include generating zero or more binary events.

[0078] In an implementation of the embodiment, the accounting for a bounded ratio of events to information pieces may include accounting for an average bounded ratio of events to information pieces.

[0079] In an implementation of the embodiment, the accounting for the average bounded ratio of events to information pieces may include tracking a number of events for each information piece.

[0080] In an implementation of the embodiment, the tracking the number of events for each information piece may include tracking the number of events for each generated information piece using a counter.

[0081] In an implementation of the embodiment, the tracking the number of events for each information piece using a counter may include tracking the number of events responsive by incrementing the counter by a predetermined incrementation value when an event is processed.

[0082] In an implementation of the embodiment, the incrementing the counter by a predetermined incrementation value may include incrementing the counter by one.

[0083] In an implementation of the embodiment, the tracking the number of events for each information piece may include subtracting the predetermined ratio value from the counter register when an information piece is processed.

[0084] In an implementation of the embodiment, the bounding the average ratio of events to information pieces may include bounding the average ratio of events to information pieces to a value of four.

[0085] In an implementation of the embodiment, the accounting for the average bounded ratio of events to information pieces may further include identifying at least one stuffing information piece within the information sequence using the average bounded ratio of events to information pieces.

[0086] In an implementation of the embodiment, the accounting for a bounded ratio of events to information pieces may include tracking a number of information pieces using a first counter, tracking a number of events using a second counter, and comparing a value of the first counter with a value of the second counter.

[0087] In an embodiment, a method of arithmetic decoding using an arithmetic decoder, to convert an information sequence including at least one information piece to an event sequence including a plurality of events, may be provided. The method of arithmetic decoding may include: receiving at least one information piece of the information sequence and a probability estimate; and generating zero or more events responsive to the information piece and the probability estimate, including computing a sub-range value for a particular event value using the probability estimate, adjusting at least one of a range register and a value register responsive to the at least one information piece and the computed sub-range value, and emitting an event responsive to a comparison between the computed sub-range value and the value register, and renormalizing a portion of the arithmetic decoder, including accounting for an average ratio of events to information pieces, by incrementing a counter register for tracking the average ratio by a predetermined incrementation value where a value of the range register is $>$ a predetermined range value, and decrementing the counter register by a predetermined decrementation value where the value of the range register is $<$ the predetermined range value.

[0088] In an implementation of the embodiment, the decrementing the counter register may include determining whether a value of the counter register is $<$ zero, and doubling the value register, adding a value of the emitted event to the value register, and decrementing the counter register by the predetermined decrementation value when the value of the counter register is not $<$ zero, and doubling the range register, doubling the value register, adding a value of the emitted event to the value register, and decrementing the counter register by the predetermined decrementation value when the value of the counter register is $<$ zero.

[0089] In an implementation of the embodiment, the receiving the at least one event may include receiving at least one binary event, and the generating zero or more information pieces may include generating zero or more binary bits.

[0090] In an embodiment, a video encoding system for encoding a video signal into an information sequence including at least one information piece, may be provided. The video encoding system may include: a video processor for converting the video signal to a sequence of events including a plurality of events representing the video signal; and an encoder in communication with the video processor for converting the sequence of events to the information sequence, including a core engine having a controller for receiving at least one

event of the event sequence and a probability estimate for the at least one event, and for generating zero or more pieces of the information sequence responsive to the received event and the probability estimate by bounding the average ratio of events to information pieces.

[0091] In an embodiment, a video decoding system for decoding an information sequence including at least one information piece into a video signal, may be provided. The video decoding system may include: a decoder for converting the information sequence to a sequence of events including a plurality of events, having a controller for receiving at least one information piece of the information sequence and a probability estimate, and for generating zero or more events of the event sequence responsive to the at least one information piece and probability sequence by accounting for an average bounded ratio of events to information pieces; and a video processor in communication with the decoder for converting the sequence of events to the video signal.

[0092] In an implementation of the embodiment, the video controller signal accounting for a bounded ratio of events to information pieces may include tracking the average ratio of events to information pieces using a counter, wherein the counter may be incremented by an incrementation value for each processed event.

[0093] In an implementation of the embodiment, the controller accounting for a bounded ratio of events to information pieces may include tracking the average ratio of events to information pieces using a first counter to track a number of information pieces, and a second counter to track a number of events, and comparing a value of the first counter with a value of the second counter.

[0094] In an implementation of the embodiment, the video signal may include a plurality of samples representing the video image, and the video processor converting the sequence of events to the video signal may include converting the sequence of events to the plurality of samples.

[0095] In an implementation of the embodiment, the video signal may include a plurality of sample blocks representing the video image, and the video processor converting the sequence of events to the video signal may include converting the sequence of events to the plurality of sample blocks.

[0096] In an implementation of the embodiment, the sample blocks may include at least one of 16x16 samples, 8x8 samples and 4x4 samples.

[0097] In an embodiment, a sequence of information pieces, generated by an arithmetic encoder from a sequence of events, including zero or more stuffing information pieces such that the ratio of events to information pieces is bounded by a predetermined ratio, may be provided.

[0098] In an implementation of the embodiment, the predetermined ratio of events to

information pieces may be four.

[0099] In an implementation of the embodiment, the sequence of information pieces may be a sequence of bits generated by an arithmetic encoder from a sequence of binary events, and including zero or more stuffing bits such that the ratio of binary events to the number of bits in the sequence is bounded by the predetermined ratio.

[0100] In an implementation of the embodiment, the information pieces and the zero or more stuffing

[0101] In an embodiment, an entropy coder for converting at least one event sequence, each event sequence including a plurality of events, to at least one information sequence, each information sequence including at least one information piece, may be provided. The entropy coder may include: an arithmetic encoder for generating zero or more information pieces of at least one information sequence responsive to at least one event of at least one event sequence; and a controller coupled with the arithmetic encoder for constraining a number of events in at least one event sequence as a function of the number of generated information pieces in at least one information sequence.

[0102] In an implementation of the embodiment, at least one event sequence may represent input data partitioned into a number of segments, and the controller constraining a number of events as a function of the number of information pieces may include: the controller constraining a maximum number of events in at least one event sequence as a function of a linear combination including a number of generated information pieces in at least one information sequence, and the number of segments.

[0103] In an implementation of the embodiment, the controller may constrain the number of events of at least one event sequence to the number of information pieces of at least one information sequence by generating zero or more stuffing pieces in at least one information sequence.

[0104] In an implementation of the embodiment, the input data may be video data.

[0105] In an implementation of the embodiment, each segment may include a block of 16 by 16 samples of the input data.

[0106] In an implementation of the embodiment, the controller may constrain the number of events of at least one event sequence to the number of information pieces of at least one information sequence by generating zero or more stuffing pieces in at least one information sequence.

[0107] In an implementation of the embodiment, at least one of the information pieces and the events may be M-ary in nature.

[0108] In an embodiment, a method for converting at least one event sequence, each event sequence including a plurality of events, to at least one information sequence, each information sequence including at least one information piece, may be provided. The method for converting at least one event sequence may include: generating zero or more information pieces of at least one information sequence responsive to at least one event of at least one event sequence; and constraining a number of events in at least one event sequence as a function of the number of generated information pieces in at least one information sequence.

[0109] In an implementation of the embodiment, at least one event sequence may represent input data partitioned into a number of segments, and the constraining may include: constraining a maximum number of events in at least one event sequence as a function of a linear combination including a number of generated information pieces in at least one information sequence, and the number of segments.

[0110] In an embodiment, an encoding apparatus for converting input data to at least one information sequence, each information sequence including at least one information piece, may be provided. The encoding apparatus may include: a processor for converting input data into at least one event sequence, each event sequence including a plurality of events; an arithmetic encoder for generating zero or more information pieces of at least one information sequence responsive to at least one event of at least one event sequence; and a controller coupled with the arithmetic encoder for constraining a number of events in at least one event sequence as a function of the number of generated information pieces in at least one information sequence.

[0111] In an implementation of the embodiment, the controller may constrain the number of events of at least one event sequence to the number of information pieces of at least one information sequence by generating zero or more stuffing pieces in at least one information sequence.

[0112] In an implementation of the embodiment, the processor may partition the input data into a number of segments, and the controller constraining a number of events as a function of the number of information pieces may include: the controller constraining a maximum number of events in at least one event sequence as a function of a linear combination including a number of generated information pieces in at least one information sequence, and the number of segments.

[0113] In an implementation of the embodiment, the arithmetic encoder and the controller may include an entropy encoder, the entropy encoder coupled with the processor, and capable of sending information regarding a number of generated information pieces in at least one information sequence to the processor.

[0114] In an implementation of the embodiment, the processor may be capable of reducing an event rate of at least one event sequence responsive to the information received from the entropy encoder.

[0115] In an implementation of the embodiment, the input data may be video data.

[0116] In an implementation of the embodiment, each segment may include a block of 16 by 16 samples.

[0117] In an implementation of the embodiment, the controller may constrain the number of events of at least one event sequence to the number of information pieces of at least one information sequence by generating zero or more stuffing pieces in at least one information sequence.

[0118] In an implementation of the embodiment, at least one of the information pieces and the events may be M-ary in nature.

[0119] In an embodiment, a sequence of information pieces, generated at an entropy encoder from at least one event sequence, each event sequence including a plurality of events, may be provided, where a number of information pieces in the sequence of information sequence may be utilized to constrain a maximum number of events of at least one event sequence as a function of a linear combination including a number of generated information pieces in the sequence of information pieces, and the number of segments.

[0120] In an implementation of the embodiment, the sequence of information pieces may include zero or more stuffing information pieces such that the constraint on the maximum number of events of at least one event sequence is satisfied.

[0121] In an implementation of the embodiment, at least one of the information pieces and the events may be M-ary in nature.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0122]

Figure 1 is a block diagram of an encoder in accordance with an embodiment of the invention;

Figure 2 is a flow chart illustrating initialization of the encoder of Figure 1 in accordance with an embodiment of the invention;

Figure 3 is a flow chart illustrating operation of the core engine of the encoder of Figure 1 in accordance with an embodiment of the invention;

Figure 4 is a flow chart illustrating the renormalization procedure for the encoder of Figure 1 in accordance with an embodiment of the invention;

Figure 5 is a flow chart illustrating the operation of the core engine of the encoder of Figure 1 upon receiving a terminate signal in accordance with an embodiment of the invention;

Figure 6 is a flow chart illustrating a Bit-Plus-Follow routine carried out by the core engine of Figure 1 in accordance with an embodiment of the invention;

Figure 7 is a block diagram illustrating a decoder in accordance with an embodiment of the invention;

Figure 8 is a flow chart illustrating initialization of the decoder of Figure 7 in accordance an embodiment of the invention;

Figure 9 is a flow chart illustrating the operation of the core engine of the decoder of Figure 7 in accordance with an embodiment of the invention;

Figure 10 is a flow chart illustrating renormalization of the core engine of the decoder of Figure 7 in accordance with an embodiment of the invention; and

Figure 11 is a block diagram of an encoder in accordance with another embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0123] In accordance with an embodiment of the invention, an arithmetic encoder is provided for converting an event sequence comprised of a plurality of events to an information sequence comprised of at least one information piece, including at least one input port for receiving events of the event sequence, and at least one output port, where a sequencer is coupled with the at least one input port and the at least one output port and includes a context generator for generating context information for at least one of the received events. A probability estimator coupled with the sequencer is provided, for receiving an event of the event sequence and corresponding context information from the sequencer, and for generating a probability estimate that the event has a particular value. A core engine is coupled with the sequencer and the probability estimator for receiving the event of the event sequence from the sequencer, and the probability estimate from the probability estimator, and generating 0 or more pieces of the information sequence responsive to the received event and the probability estimate. The core engine may bound the ratio of events to information pieces, and provides the 0 or more information pieces of the information sequence to the sequencer, where the generated information pieces are provided as the information sequence at the at least one output port.

[0124] In accordance with another embodiment of the invention, an arithmetic decoder is also provided for converting an information sequence comprised of at least one information piece to an event sequence comprised of a plurality of events, including at least one input port for receiving information pieces of the information sequence, at least one output port, and a sequencer coupled with the at least one input port and the at least one output port and

including a context generator for generating context information for at least one generated event. The probability estimator is coupled with the sequencer for receiving a generated event of the event sequence and the corresponding context information, and for determining a probability estimate. A core engine is coupled with the sequencer and the probability estimator for processing zero or more information pieces of the information sequence from the sequencer responsive to the probability estimate to generate one event by accounting for a bounded ratio of events to information pieces. The generated event is provided to the probability estimator with corresponding context information from the sequencer for updating the probability estimate, and the generated event is provided to the sequencer for transmittal as the event sequence from the at least one output port.

[0125] Having the arithmetic encoder which encodes the event sequence to the information sequence by bounding the ratio of events to information pieces allows for compression of a sequence of data (i.e. binary or M-ary events) in a fashion which allows real-time use of the event sequence. Further, reducing the complexity of the generated information sequence by bounding the ratio of events to information pieces allows decoders with limited processing capabilities (e.g. mobile decoders) to decode the information sequence back to its corresponding event sequence while potentially maintaining real-time processing capabilities, or at least without requiring a user of the information to wait for a significant amount of time while such decoders perform the decoding operations.

[0126] Similarly, providing a decoder which decodes the information sequence to an event sequence by accounting for a bounded ratio of events per information piece provides for such real-time use of the event sequence, for example during a video conference call, or for the retrieval and processing of information from storage media, for example retrieving video information such as a movie from a DVD-ROM and processing the video data for use.

[0127] In accordance with another embodiment of the invention, an entropy coder for converting at least one event sequence, each event sequence including a plurality of events, to at least one information sequence, each information sequence including at least one information piece, is provided, including an arithmetic encoder for generating zero or more information pieces of at least one information sequence responsive to at least one event of at least one event sequence. The entropy encoder further includes a controller coupled with the arithmetic encoder for constraining a number of events in at least one event sequence as a function of the number of generated information pieces in at least one information sequence. The at least one event sequence may represent input data partitioned into a number of segments, where the controller constrains a number of events as a function of the number of information pieces by constraining a maximum number of events in at least one event sequence as a function of a linear combination including a number of generated information pieces in at least one information sequence, and the number of segments. The entropy encoder may be coupled with a processor for converting input data into the at least one sequence of events, where the entropy encoder is capable of sending information regarding a number of generated information pieces in at least one information sequence to the processor. Further, the processor may be capable of reducing an event rate of at least one event

sequence responsive to the information received from the entropy encoder.

[0128] Having an entropy encoder capable of constraining a number of events in an event sequence as a function of the number of generated information pieces in an information sequence may allow for the video data to be consistently decoded, accounting for a maximum number of information pieces per group of segments, while maintaining video encoding quality as compared with encoders of the prior art. Additionally, where the encoder receives an indication that the complexity threshold is about to be exceeded, the encoder may generate less events for the next segment of the input data, for example reducing a chance that the constraint threshold for the next segment is exceeded. Further, as the controller is capable of triggering the addition of stuffing bits where the constraint threshold is exceeded, the processor may be implemented in a way that the processor need not account for, or deal with, the complexity constraint. The encoders and the decoder may be utilized in the processing of video data, for example as generated by a video processor (i.e. video codec). In the video processor example, a video image is recorded, and partitioned into sample blocks of data which may represent 16x16, 8x8, or 4x4 samples of the recorded image. The blocks are then transformed by the video processor (i.e. utilizing a discrete cosine transform), and quantized to yield integer values representing the sample block. The integer values are converted to a sequence of events (i.e. binary events) by the video processor, and sent to the encoder for encoding. Alternatively, the video processor may operate directly on individual samples, transforming and quantizing them, and converting the particular quantized integer value for the sample to a sequence of events.

[0129] Although one such use for the encoder and decoder discussed herein is in the encoding and decoding of video data, one skilled in the art would realize that the encoder and decoder described herein may be utilized in any situation where a sequence of events is compressed to an information sequence in the case of the encoder, and where such an information sequence is decompressed in the case of the decoder. Further, although the following discussion of the encoder is in the context of processing a sequence of events comprising a plurality of binary events to an information sequence comprising at least one information binary bit, and for the decoder is in the context of processing a sequence of information comprising at least one information binary bit to a sequence of events comprising a plurality of binary events, that the encoder and decoder could operate on event sequences and information sequences comprised of events which are M-ary in nature (i.e. each M-ary event represents >1 bits of data) using the teachings described herein, as would be appreciated by one skilled in the art.

[0130] Figure 1 illustrates a block diagram of an arithmetic encoder 100 in accordance with an embodiment of the invention. The arithmetic encoder 100 includes a sequencer 105, a probability estimator 110, and a core engine 115, which are each coupled with one another. One or more input data lines 120 provide an input port for receiving a sequence of events 125, here an ordered sequence of binary events, to the encoder 100. The binary events of the sequence 125 may have a value "A" or "B". The event sequence is processed by the encoder, as described below, to generate an information sequence which is an ordered sequence

comprised of at least one information piece, here binary bits, where the number of information pieces in the information sequence is less than the number of events in the event sequence. The output data lines 130 provide an output port for sending the information sequence 135 from the encoder 100. The ordered sequence of binary bits of the information sequence includes one or more bits having a value of "0" or "1".

[0131] Upon receiving the ordered sequence of binary events 125, the sequencer 105 sequentially transmits the events 125 to both the probability estimator 110 and the core engine 115 via event transmission lines 140 and 145 respectively. For each binary event of the event sequence 125, the sequencer 105 also transmits context information to the probability estimator 110 for the binary event via a context transmission line 150. The probability estimator 110, using the context information received via the context transmission lines 150, generates a probability estimate $P(A)$ which is transmitted to the core engine 115 via probability transmission line 155. The probability estimator 110 then updates its internal state based on the value of the binary event received via binary event transmission line 140. The core engine 115 produces 0 or more information bits using the binary event received via event transmission line 145 and the corresponding probability estimate $P(A)$ received via probability transmission line 155, while bounding a ratio of events to information bits, as will be described in more detail below.

[0132] The sequencer 105, upon receipt of a first binary event of the event sequence 125, sends an initialization signal to the probability estimator 110 and the core engine 115 via initialization lines 185 and 190 respectively.

[0133] In producing the zero or more information bits for transmission, through bit transmission line 158, the core engine 115 utilizes various registers including a range register 165, a low register 170, a bits outstanding register 175, and a counter register 180. Operation of the encoder 100 is described with respect to the flowcharts of Figures 2-6.

[0134] Figure 2 illustrates a flowchart describing operation of the core engine 115 upon receipts of an initialization signal from the sequencer 105. As shown in the flowchart of Figure 2, the core engine 115 initializes the range register 165, for example to a value 8000 hexadecimal (H), as shown at step 205. The core engine 115 also initializes the low register 170, the bits outstanding register 175 and the counter register 180 each to, for example, a value of 0, as shown at steps 210, 215 and 220 respectively.

[0135] After initialization, and upon receiving a binary event of the event sequence 125, the core engine operates as illustrated in the flowchart of Figure 3 in accordance with an embodiment of the invention.

[0136] As shown at step 305 of Figure 3, the core engine 115 computes a sub-range value $R(A)$ for a particular event value, for example an event value "A", using a current value of the range register 165 and the probability estimate for that particular event value received via the probability transmission line 155. Here, the sub-range value $R(A)$ is computed by multiplying

the range register value "R" by the probability estimate $P(A)$. As shown at step 310, a comparator of the core engine 115 (not shown) compares a value "X" of the binary event received via event transmission line 145 with the particular event value A for which the probability $P(A)$ has been generated. Where the received event value X is equal to A, the range register 165 is set to the sub-range value $R(A)$, as shown at step 315 (i.e. the value of $R(A)$ is stored into the range register 165). However, where the receive event value X is not equal to the event A at step 310, the sub-range $R(A)$ is subtracted from the value R of the range register 165, where the resulting value is stored in the range register 165 as shown at step 320. A value L of the low register 170 is summed with the sub-range value $R(A)$ and stored in the low register 170 as shown at step 325. Upon recomputing the value of the range register 165 at step 315, or the low register 170 at step 325, the core engine calls a renormalization routine, as shown at step 330 and further discussed below with respect to the flow chart of Figure 4, in accordance with an embodiment of the invention.

[0137] As shown in Figure 4, renormalization begins where a value of the range register 165 is compared with a predetermined range value, here 4000H as shown at step 405. Where the value R of the range register 165 is $> 4000H$, the counter register is incremented by 1, a predetermined incrementation value, here as shown at step 410. Where it is determined at the core engine 115 that the value R is less than or equal to 4000H at step 405, the value L of the low register 170 and R of the range register 165 are summed, and compared with the value 8000H as shown at step 415. Where the sum is less than or equal to 8000H, a bit plus follow routine is called having a parameter B equal to "0" as shown at step 420 and as will be further discussed below. The value of the low register is then doubled as shown at step 425, and the value C of the counter register 180 is compared with 0 as shown at step 430. Where the value C of the counter register 180 is not less than 0, a decrementation value, here 4H, is subtracted from the counter register 180 as shown at step 435, and flow returns to step 405 and operates as described above. However, where it is determined at step 430 that the value C of the counter register is less than 0, the range register is rescaled by doubling the value R of the range register 165 as shown at step 440, and program flow continues to step 435 as discussed above.

[0138] Where it is determined at step 415 that the sum of the values L and R is not less than or equal to 8000H, the value L of the low register 170 is compared with 8000H as shown at step 445. Where the value L is greater than or equal to 8000H, the bit plus follow routine is called with a parameter B equal to "1" as shown at step 450 and discussed further below. A value 8000H is subtracted from the low register 170 as shown at step 455, and flow continues to step 425 as discussed above. However, where it is determined that L is not greater than or equal to 8000H at step 445, the bits outstanding register 175 is incremented by 1 as shown at step 460, and a value 4000H is subtracted from the low register 170 as shown at step 465. Flow then continues to step 425 as discussed above.

[0139] Upon receipt of a final event of the event sequence 125, the sequencer 105 sends a terminate signal to the core engine 115 via terminate transmission line 195. Operation of the core engine 115 when a terminate signal is received is shown in the flow chart of Figure 5 in

accordance with an embodiment of the invention.

[0140] As shown in Figure 5, a bit plus follow routine is called, step 505, with an argument B having the value of the most significant bit (MSB) of the low register 170. The least significant bits (LSB) of the low register 170 are then sent to the sequencer 105 as shown at step 510. The bit plus follow routine is shown in more detail in the flow chart of Figure 6 in accordance with an embodiment of the invention.

[0141] As shown in Figure 6, a value B is sent as a bit of the information sequence 135 to the sequencer 105 via bit transmission line 158, as shown at step 605. A value BO of the bits outstanding register 175 is compared with 0, at step 610. Where a value BO of the bits outstanding register 175 is not greater than 0, flow returns to whichever program segment/routine that called the bit plus follow routine (i.e. box 420 or box 450 of Figure 4, or box 505 of Figure 5). However, where the value BO is greater than 0 at step 610, a bit of the information sequence 135 having a value of 1 less the value B is sent to the sequencer 105 via bit transmission line 158, as shown at step 615. The value BO of the bits outstanding register is decremented by 1 as shown at step 620, and flow returns to step 610, and continues as discussed above.

[0142] Thus, providing the encoder 100 having a core engine 115 with the counter 180 allows for tracking the ratio of events to information pieces, thereby allowing such ratio to be bounded in order to reduce the complexity bound of the information sequence provided by the encoder. The counter incrementation value discussed with respect to step 410, the counter decrementation value discussed with respect to step 435 and the value which the counter register 180 is compared against at step 430 are used in governing the average ratio of events to information pieces. For example, with the values discussed above with respect to Figure 4, having a decrementation value of 4 at step 435 results in the average ratio of events to information pieces being bounded at 4. Altering the decrementation value alters the average ratio of events to information pieces. Accordingly, the incrementation value, the decrementation value and the value to which the counter is compared are merely exemplary, as such values, along with the value to which the counter register 180 has been initialized at box 220 of Figure 2, need not be as described above, but may be any integer value, where the corresponding values will be selected to achieve the desired bounded average ratio of events to information pieces as would be appreciated by one skilled in the art.

[0143] Figures 7- 10 illustrate an arithmetic decoder and the operation thereof which is capable of decoding an information sequence comprised of at least one information piece to an event sequence comprising a plurality of events, by accounting for a bounded ratio of events to information pieces in the information sequence. Figure 7 is a block diagram of the decoder, where Figures 8-10 are flowcharts illustrating operation of the decoder.

[0144] Figure 7 illustrates an arithmetic decoder 700 in accordance with an embodiment of the invention. The decoder 700 includes a sequencer 705, a probability estimator 710, and a core engine 715 coupled with one another. One or more information sequence input data lines 720

provide an import port for a sequence of information 725, here in ordered sequence of binary bits, to the decoder 700. The binary bits of the sequence 725 may have a value of "0" or "1". The decoder processes the information sequence to generate an event sequence by accounting for the bounded ratio of events to information pieces in the information sequence, as discussed further below. The generated sequence of events is an ordered event sequence comprising a plurality of events, here binary events, which may have a value of "A" or "B". The event sequence is provided to output event sequence data lines 730, which provides at least one output port from the decoder 700.

[0145] Upon receiving the ordered sequence of binary bits 725, the sequencer 705 sequentially transmits the one or more bits to the core engine 715 via bit transmission line 740. Upon receiving the one or more bits, the core engine 715 begins generating events, here binary events, which are transmitted to the sequencer 705 and probability estimator 710 via event transmission lines 745 and 750 respectively. As further discussed below, the probability estimator sends an initial probability estimate to the core engine for generating the first binary event. Thereafter, for each binary event generated by the core engine 715 and sent to the sequencer 705, the sequencer 705 transmits a corresponding context to the probability estimator 210 over the context transmission line 755. Based on the value of the context received via context transmission line 755, the probability estimator 710 generates a corresponding probability estimate $P(A)$, which is sent to the core engine 715 over probability transmission line 760, and used by the core engine 715 in generating further events. After transmitting the probability estimate $P(A)$, the probability estimator 710 updates its internal state based on the value of the binary event received from the core engine 715 over event transmission line 750. The core engine 715 consumes zero or more information bits for each binary event generated. The core engine 715 utilizes various registers in generating the events of the event sequence 735, including a range register 765, a value register 770 and a counter register 775. An initialization signal is provided to the core engine 715 and the probability estimator 710 via initialization transmission lines 780 and 785 respectively. A terminate signal is provided to the core engine 715 via a terminate signal line 790. Operation of the decoder 700 is shown in the flowcharts of Figures 8-10.

[0146] Figure 8 illustrates operation of the decoder 700 upon receipt of an initialization signal via initialization signal line 780 in accordance with an embodiment of the invention. As shown in Figure 8, the core engine 715 initializes the range register 765 to a value of 8000H as shown at step 800, and initializes the value register 770 to a particular value represented by 16 bits (discussed below) as shown at step 805. Further, the core engine 715 initializes the counter register 775 to a value of zero as shown at step 810.

[0147] The value register 770 is initialized at step 805 by collecting 16 bits into the value register 770, for example the first 16 information pieces of the information sequence, as described in Arithmetic Coding Revisited, by Moffat et al. , ACM Transactions on Information Systems, VOL. 16, NO. 3, July 1998, pages 256-294.

[0148] Figure 9 is a flow chart illustrating operation of the core engines 715 in generating the

event sequence 725 in accordance with an embodiment of the invention. As shown at step 900, a subrange value $R(A)$ is computed by the core engine 715 for an event value "A". The subrange value $R(A)$ is computed based on the value of the range register 765 and determined probability estimate $P(A)$ received via the probability estimate transmission line 760 from the probability estimator 710, by multiplying the value R of the range register 765 by the probability estimate $P(A)$. When computing the sub-range value $R(A)$ for the first time for a particular information sequence 735, the probability estimator 710 determines the probability estimate $P(A)$ to be some initial/predetermined probability estimate (i.e. agreed-upon ahead of time). Thereafter, as events are emitted from the core engine 715, the probability estimator 710 utilizes the emitted events along with corresponding context information received via context information line 755 in updating the probability estimate $P(A)$ provided via probability estimate transmission line 760, as discussed below.

[0149] The subrange value $R(A)$ is then compared with a Value V of the value register 770 as shown at step 905. Where the subrange value $R(A)$ is less than the value V of the value register 765, the range register 165 is set to the subrange value $R(A)$ as shown at step 910, and an event "A" is emitted to the sequencer 705 via event transmission line 745 as shown at step 915. A renormalization procedure is then called, as shown at step 920, and which will be discussed in further detail with respect to the flow chart of Figure 10.

[0150] Where the subrange value $R(A)$ is not less than the value V of the value register 770 at step 905, the subrange value $R(A)$ is subtracted from the range register 765 as shown at step 925. The subrange value $R(A)$ is subtracted from the value register 770 as shown at step 930, and an event "B" is emitted from the core engine to the sequencer 705 via event transmission line 745, as shown at step 935.

[0151] Flow continues to the renormalization procedure at step 920.

[0152] As discussed above, upon emission of an event via the event transmission line 745, the sequencer 705 transmits a context via the context transmission line 755 to the probability estimator 210, where the probability estimator 710 updates its internal state based on the value of the binary event emitted by the core engine 715.

[0153] Figure 10 is a flow chart illustrating operation of the core engine 115 when carrying out the renormalization discussed above with respect to step 920 in accordance with an embodiment of the invention. As shown at step 1000, the value R of the range register 765 is compared with a predetermined range value, here 4000H. Where the predetermined value R is not less than or equal to 4000H, the counter register 775 is incremented by a predetermined incrementation value, here 1, as shown at step 1005, and renormalization is complete. However, where the value R of the range register 765 is less than or equal to 4000H a value C of the counter register 775 is compared with 0 as shown at step 1010. Where the value C of the counter register 775 is not less than 0, the value V of the value register 770 is doubled, and added with a value of the most recent bit received at the core engine 715 via the bit transmission line 740, where the resulting sum is stored in the value register 770 as shown at

step 1015. As shown at step 1020, a value C of the counter register 775 is decremented a predetermined decrementation value, here 4H, and flow returns to step 1000. Where a value C of the counter register is less than 0 at step 1010, the range register 765 is rescaled by doubling the value R of the range register as shown at step 1025, and flow continues to step 1015 as described above.

[0154] After a final piece of information of the information sequence 725 is transmitted to, and processed by the core engine 715, the sequencer 705 transmits a terminate signal to the core engine 715 via terminate signal transmission lines 790 indicating to the core engine 715 that the decoding of the information sequence 725 is complete. The core engine 715 need perform no operation in response to the termination signal.

[0155] The encoder 100 with a reduced complexity bound that is capable of encoding an event sequence to an information sequence by bounding a ratio of events to information pieces is advantageous as providing an information sequence which may be decoded and used to support the providing of the information for real-time use, without significantly affecting the encoding efficiency of the information sequence. Additionally, such a reduced-complexity encoding may be beneficial where the decoder used in decoding the information sequence possesses reduced-processing capabilities (i.e. is on a mobile device) as such a decoder would be capable of decoding the information sequence without undue delay in waiting for the decoding to be accomplished. Further, the decoder 700 capable of decoding an information sequence to an event sequence by accounting for a bounded ratio of events to information pieces in the information sequence, may be advantageous as allowing an efficient decoding of the information sequence for use in, for example, real-time use of the event sequence.

[0156] Figure 11 illustrates an encoder 1100 in accordance with another embodiment of the invention. As shown in Figure 11, the encoder includes a processor 1105 coupled with an entropy encoder 1110. The entropy encoder includes an arithmetic encoder 1115 coupled with a controller 1120. Input data 1125 is received at the encoder, and is encoded to one or more sequences of information pieces shown at 1130.

[0157] The input data 1125 may be any data to be encoded by the encoder 1100, including, for example, video data. The processor 1105 receives the input data, and converts the input data into one or more sequences of events 1135. The processor 1105 may partition the input data into one or more segments of information (not shown). For example, where the input data 1125 is video data, the processor may partition the video data into blocks of data, for example representing 16x16 samples of the corresponding video image. A sequence of events 1135 may represent one or more segments (video blocks).

[0158] The entropy encoder 1110 receives the sequence of events 1135 to be arithmetically encoded, where the arithmetic encoder 1115 generates 0 or more bits of the information sequence 1130 for each event of the sequence of events 1135. The controller 1120 is capable of regulating the entropy coder 1110 by constraining the number of information pieces of the information sequence 1130 as a function of the number of the events of the sequence of

events 1135, such that the number of events does not exceed a threshold with respect to the number of information pieces generated. Where the threshold is exceeded, the controller 1120 may control the entropy coder to add one or more stuffing pieces of information to the information sequence 1130 to achieve the threshold.

[0159] The entropy encoder may encode the sequence of events 1135 in a similar fashion as described above with respect to the arithmetic encoder 100 of Figure 1. For example, the arithmetic encoder 1115 may include, where appropriate, a sequencer, and probability estimator as described above with respect to Figure 1. The arithmetic encoder 1115 may further include a core engine with Range, Low and Bit Outstanding Registers. The controller 1120 may include one or more counters for tracking a number of events of the sequence of events 1135 processed and a number of information pieces of the information sequence 1130 generated. Alternatively, the one or more counters may be present within the entropy coder 1110, where the controller 1120 is capable of monitoring the counter(s) to ensure the threshold between events processed and information pieces generated is substantially maintained. The controller 1120 may constrain the number of information pieces, for example, over each individual sequence of events being encoded. For example, the controller may ensure that $(\# \text{ generated information pieces}) \times (\text{threshold}) \geq (\# \text{ events processed})$. This may be accomplished, for example, in a similar fashion as discussed above with respect to Figures 1-6, and specifically Figure 4, where the controller increments a counter for each event processed, and subtracts a number from the counter representing the threshold value, for example "4" when a bit is processed. The controller may then compare the value of the counter, with for example the value 0, and trigger the insertion of a stuffing bit if the value of the counter is larger than 0. In this way, the controller is capable of ensuring that the arithmetic encoder 1115 is substantially constrained to a 4:1 ratio of events processed to information pieces generated.

[0160] In addition, or in the alternative, the controller may constrain the number of events of the sequence of events 1135 as a function of the number of information pieces in the sequence of information pieces, and a number of segments of the input data 1125 represented in the sequence of events. For example, the constraining may take the form of a linear combination:

$$e \leq \alpha B + \beta S,$$

where

e is the number of events represented in the sequence of information pieces,

B is a number of information pieces in the sequence of information pieces,

S is a number of segments represented in the sequence of events, and

α and β represent a decrement value to a counter utilized by the controller to substantially maintain a constraint of the number of events of the sequence of events with respect to a number of information pieces generated.

[0161] The values for α and β are typically provided to the controller 1120, and the derivation of α and β will be discussed below. The value α may represent a decrement value to, for example, a counter upon generation of an information piece at the coder, where the value β may represent decrement value to, for example, a counter upon completion of processing a segment. In the alternative, the value β may be decremented from a counter value at the beginning of processing of a segment, or at any other time during the processing of a segment as would be apparent to one skilled in the art.

[0162] A rate control line 1140 may be utilized in constraining the number of events of the sequence of events 1135 as a function of the number of information pieces in the sequence of information pieces, and a number of segments of the input data 1125 represented in the sequence of events. For example, the rate control line 1140 may be used to carry information relating to the complexity constraint (MEPS and/or PMEPS, see below) from the entropy encoder 1110 to the processor 1105. The entropy encoder may indicate to the processor that the constraint threshold is about to be, or is exceeded, for example, by transmitting information regarding the value of the counter(s). The processor 1105 may thus reduce the event rate (i.e. reduce the number of events per segment/block.), for example, to reduce the likelihood of the controller 1120 triggering the insertion of stuffing bits into the information sequence 1130.

[0163] In some circumstances, the particular number of segments of data of the input data 1125 that will be processed may be known at the encoder 1100, for example, where the sequence of information pieces need not be substantially limited for transmission over a particular communication medium. In these circumstances, the sequence of information pieces generated at the encoder 1100 may represent an entire coded image for transmission generated from a known number of segments of the input data 1125. Where the particular number of segments of data of the input data that will be processed is known, the controller may, for example, constrain the number of events in the sequence of events 1135 as:

$$e - \beta S \leq \alpha B,$$

Since the total number of segments, S , and the value β are known, the product of $\beta \times S$ may be subtracted from the number of events, e , for the sequence of events 1135 during or after processing of the one or more segments of the input data 1125. For example, where a counter is utilized to constrain the number of events responsive to the number of bits that have been generated, the counter may initially be decremented by a value of $\beta \times S$, and may be decremented by a value α for each information piece generated, while the counter is incremented by "1" for each event of the sequence of events 1135 processed by the entropy encoder 1110. The value of β may be any value, typically in the range of 1 to 100, and may be determined, for example, as described further below. The value of α may be any value, typically in the range of 1 to 10, and may be determined, for example, as described further below.

[0164] In some circumstances, a number of segments of the input data 1125 to be processed is not known in advance, for example, where the communication medium limits the number of information pieces that may be provided in the information sequence 1130. This may occur, for

example, where the information sequence 1130 is to be transmitted over the Internet, as an Internet Protocol (IP) packet, where the IP packet has a maximum size limitation. In these circumstances, depending on the complexity of a particular image, one or more sequences of information pieces may be required to represent a single image of the input data 1125. However, the number of segments utilized for generation of a sequence of information pieces may not be known in advance, since it may not be known after how many processed segments the maximum size of a sequence of information pieces will be reached. Where a number of segments of the input data 1125 to be processed is not known in advance, the controller may account for the plurality of sequences of events as the one or more segments representing a particular sequence of events is encoded. For example, where a counter is utilized to constrain the number of events responsive to the number of bits that have been generated, the counter may be decremented by a value β for each segment processed, and may be decremented by a value α for each information piece generated, while the counter may be incremented by "1" for each event of the sequence of events 1135 processed by the entropy encoder 1110.

[0165] The values for α and β may be determined, for example, by accounting for limitations imposed by a particular standard (i.e. a standard governing video coding), and/or limitations with respect to a decoder that will be utilized to decode the information sequence encoded by the encoder 1100. Such limitations imposed by the particular standard may include information regarding a maximum number of bits per second (MBPS), a maximum number of segments (blocks) per second (MSPS), and a peak maximum number of bits per second (PMBPS). The PMBPS may represent, for example, a maximum number of bits of a picture divided by the temporal distance between two pictures to be coded.

[0166] The MBPS may account for channel capacity, for example, approximately 64 000 bits per second for mobile communication applications, 500 000 bits per second for digital subscriber line (DSL), and 4 000 000 bits per second for broadcast applications such as television (TV). The MSPS may account for the video format standard, for example, 176x144 pixels at 10 or 15 frames per second for mobile (i.e. cellular telephone) applications, 320x240 pixels at 24 frames per second for DSL applications, 720x480 pixels at 30 frames per second for standard TV applications, and 1920x1088 pixels at 30 frames per second for High-Definition TV, PMBPS may be any multiple of the MBPS, typically several times MBPS, for example, PMBPS $\approx 10 \times$ MBPS.

[0167] Depending on the particular implementation, the resources of a decoder (i.e. processing capabilities and/or available buffer memory) may be limited, and may also be taken into account when determining the values of α and β . Such limitations may include actual or assumed values for a maximum number of events per second (MEPS) that can be processed by the decoder, and/or a peak maximum number of events per second (PMEPS) that can be processed by the decoder (i.e. a maximum number of events in a picture divided by the temporal distance between two pictures to be decoded).

[0168] Accounting for one or more of the above limitations of the decoder, α and β may be represented in a linear system as:

$$\alpha + \beta = 1$$

$$PMBPS = \alpha * PMBPS + \beta * PMBPS,$$

and

$$MEPS = \alpha * MBPS + \beta * MSPS.$$

[0169] The values of α and β may be determined by solving the linear system for α and β , given for example, the limitations PMBPS and MSPS set by the particular standard, and one or both of the limitations PMEPS and MEPS of the decoder. The values for α and β need not be integer values, but rather may be any values to substantially satisfy the linear system.

[0170] The values of α and β may be determined in advance, by a system designer of the encoder 1100 accounting for one or more of the limitations discussed above, and provided to the controller 1120. Alternatively, or in addition, the values of α and β may be determined by the controller 1120, or any other component of the encoder 1100, in accordance with one or more of the limitations discussed above, or as encoder 1100 default values. Where the controller 1120 determines values for α and β using one or both of the limitations imposed by the standard or by a decoding device, the information regarding one or more of the limitations may be stored at a memory (not shown) of the controller 1120, and used by the controller 1120 in determining the α and β values. Additionally, or in the alternative, information regarding the limitations may be provided to the controller 1120, for example, by some external device such as an external memory (i.e. a Digital Video Disc (DVD)), a DVD player device, or from a systems engineer, for example, handling some of the functions with regards to encoding the particular input data 1125. In the latter case, the systems engineer may enter into a console or other input device (not shown), or otherwise specify, information with regards to limitations imposed as a result of an encoding standard and/or a decoding device, as would be appreciated by one skilled in the art.

[0171] Where one or both of the decoder limitations PMEPS and MEPS are not known, values for the PMEPS and MEPS may be assumed, for example, as minimum operational requirements for a decoder to possess to be capable of decoding the information sequence 1130 generated by the encoder 1100. Further, when assuming the values for PMEPS and/or MEPS, considerations as to, for example, a percentage of time the decoder will utilize for arithmetically decoding data, may be accounted for. For example, it may be assumed that the decoder will be capable of expending no more than 30% of its processing time/capabilities to carry-out arithmetic decoding of an information sequence received at the decoder.

[0172] Further, even though the values of α and β may be determined utilizing the above linear system, such values may be adjusted to achieve desired encoding performance, where the adjusted values for α and β may subsequently be utilized by the encoder 1100 for encoding the input data 1125 to the information sequence 1130. For example, the values of α and β may be adjusted with considerations as to a potential loss of video quality due to the information pieces stuffed in the information sequence 1130. It may be desirable, for example, to minimize the number of stuffed information pieces such as to maintain good video quality (i.e. video with little or no distortion) while bounding the decoding complexity. Such a trade-off between decoding complexity and video quality may be determined experimentally, and may account for

the particular video coding algorithm utilized by the encoder 1100.

[0173] In addition, when determining the values for α and β , considerations may be made as to whether the complexity constraint is too tight, for example, the values for α and/or β are too low. A high proportion of stuffing information pieces in the sequence of information pieces (i.e., a number of stuffing pieces greater than approximately 1% or 2% of the information pieces of the information sequence) may indicate that the constraint is too tight. One skilled would realize that other proportions may indicate a high proportion of stuffing information pieces, for example, taking into account the particular standard and/or decoder that may be used.

[0174] Where it is determined, for example, that the values for α and β are too tight, the values for α and β may be increased to reduce the likelihood that stuffing bits will be added (i.e. reducing the likelihood of a quality penalty in the encoded information sequence). When increasing the values for α and β , considerations may be made as to the effect on the resulting complexity bounds (i.e., MEPS and/or PMEPS) with respect to a decoder that will be utilized to decode the encoded information sequence. Such considerations may include the cost to implement the decoder. If the complexity bound is higher, more processing power may be required in the decoder. An increase in required processing power could likely result in higher implementation cost.

[0175] Alternatively, α and β can be determined experimentally, using linear regression techniques. A number of sequences of events, each representing S segments, may be encoded without enforcing any complexity constraint. For each sequence z of events, it is known for the number of events $e(z)$, the number of resulting generated information pieces $B(z)$. It may be determined, using linear regression, a line $e = c*B + d$ that approximates the data pairs $(e(z), B(z))$. An initial value of α may be indicated by c , and an initial value of β may be indicated by d/S . The values for α and/or β may then be increased such as to minimize the number of data pairs $(e(z), B(z))$ that lie above the line $e = \alpha*B + \beta*S$. The amount by which α and/or β are increased will likely also take the resulting values of MEPS and/or PMEPS into account, for example, to control implementation cost of a decoder.

[0176] Utilizing the values for α and β as determined by one or more of the various techniques discussed above, the encoder 1100 may account for a value of α (i.e. decrement a counter by the value of α) for each information piece generated, and may account for a value of β (i.e. decrement a counter by the value of β) upon completion of a segment of the input data 1125. For example, where α and β are integer values, such accounting (i.e. decrements to one or more counters) may be accomplished directly.

[0177] Where, for example, one or both of α and β are fractional values, a common denominator may be determined to provide non-fractional values for α and β . In this circumstance, the newly, non-fractional values for α and β may be accounted for as described above, for example by decrementing a counter by the values of α and β upon information piece generation and completion of segment processing, respectively. The determined common denominator may be accounted for, for example, by adding the value of the common

denominator to the counter value upon processing of each event of the event sequence 1135. For example, where the values for α and β are determined to be $4/3$ and 25 respectively, a common denominator may be determined as 3. Non-fractional values for α and β may thus be determined as 4 and 75 respectively, using the common denominator. Thus, where a counter is utilized to account for the values of α and β , the counter may be decremented by 4 for each information piece generated, decremented by 75 upon completion of processing each segment, and incremented by 3 for each event processed.

[0178] Having an encoder capable of constraining a number of information pieces over one or more segments of the input data may allow for the video data to be consistently decoded, accounting for a maximum number of information pieces per group of segments, while maintaining video encoding quality as compared with encoders of the prior art. Additionally, where the encoder receives a signal through the rate control line that the complexity threshold is about to be exceeded, the encoder may generate less events, for example β or less events from the next segment of the input data, to avoid exceeding that threshold for this next segment/block. Having a value of β different from zero may thus help the processor 1105 in adjusting video quality. In addition, as the controller typically triggers the addition of stuffing bits where the constraint threshold is exceeded, the processor may be implemented in a way that the processor need not account for, or deal with, the complexity constraint. Thus, if the processor satisfies the limits imposed by the particular standard, for example, limits regarding one or more of MSPS, MBPS and PMBPS, it will also satisfy the complexity limits MEPS and/or PMEPS (i.e. derived from α and β as discussed above).

[0179] Although the constraining the number of events responsive to the number of information pieces generated has been discussed with respect to the encoder 1100, one skilled will realize that a similar constraining of the number of events of the sequence of events responsive to the number of information pieces in the information sequence may be accounted for in the encoder 100 discussed above with respect to Figure 1. Further, although the controller 1120 has been described as coupled to the arithmetic encoder 1115, one skilled will realize that such coupling may be direct, or through other intervening devices. Such coupling allows direct and/or indirect communication between the controller 1120 and the arithmetic encoder 1115, such that the controller 1120 is capable of regulating the entropy coder 1110 by constraining the number of information pieces of the information sequence 1130 as a function of the number of the events of the sequence of events 1135.

[0180] The encoders 100 and 1100, and decoder 700, described herein may be comprised completely of hardware, as software running on a suitable microprocessor, or as a combination of hardware or software. The functionality of core engines 115 and 715, as well as the arithmetic encoder 1115, controller 1120 and processor 1105, may be carried out as a computer program running on a suitable microprocessor, and the counter registers 180 and 775 may be a register residing within the microprocessor, or may be resident on some external storage media in communication with the microprocessor, for example, RAM, EEPROM, or any rewriteable magnetic, optical or other storage media as would be appreciated by one skilled in the art. As discussed above, although the encoders 100 and 1100, and decoder 700, have

been discussed as processing a binary event sequence to a binary information sequence in the case of the encoders, and a binary information sequence to a binary event sequence in the case of the decoder, one skilled in the art would realize that, using the teachings described herein, and M-ary event sequence may be processed and produced while still achieving the advantages discussed herein.

[0181] Further, the sign of the counter registers 180 and 775 may be inverted while still achieving the advantages discussed herein, where the incrementations to the counter register described at steps 410 and 1005 would be changed to decrementsations, the decrementsations to the counter register described at steps 435 and 1020 would be changed to incrementations, and the comparisons with the counter register at steps 430 and 1010 would be changed to determining whether the value of the respective counter register is $> \text{zero}$.

[0182] Additionally two counters may be used instead of one. The first counter C1 may count events, and the second counter C2 may count bits. In such a case, a comparison between the two counters may be whether $C1 < 4 * C2$ in determining whether (or accounting for) the predetermined ratio of events to bits is substantially maintained.

[0183] Additionally, although the encoder 100 and decoder 700 have been described as separate hardware, one skilled in the art would realize that the functionality carried out by each as described herein could be integrated into a single unit, for example which may be useful in the case of a personal computer (PC) being used for a video conference where the processor and storage media of the PC would be used to accomplish the functionality of the encoder and decoder described herein.

[0184] The bounding ratio of events to information pieces and accounting for a bounded ratio of events to information pieces accomplished by the core engines discussed herein may be accomplished on other core engine designs, including that of a Q-coder, MQ-coder, Z-coder, as well as those discussed in "Low Complexity Arithmetic Coding Implementation", Rene van der Vleuten, Document JVT-B033, JVT meeting in Geneva, Switzerland, January 2002, and "Low-Complexity Arithmetic Coding Engine", Winger, Document JVT-B036, JVT meeting in Geneva, Switzerland, January 2002.

[0185] Thus, an encoder with a reduced complexity bound has been described which is capable of encoding an event sequence to an information sequence by bounding a ratio of events to information pieces. Such an encoder is advantageous as providing an information sequence which may be decoded and used to support the providing of the information for real-time use, without significantly affecting the encoding efficiency of the information sequence. Additionally, such a reduced-complexity encoding may be beneficial where the decoder used in decoding the information sequence possesses reduced-processing capabilities (i.e. is on a mobile device) as such a decoder would be capable of decoding the information sequence without undue delay in waiting for the decoding to be accomplished. Additionally, an encoder capable of constraining a number of information pieces over one or more segments of the input data may be described, that may allow for the video data to be consistently decoded,

accounting for a maximum number of information pieces per group of segments, while maintaining video encoding quality as compared with encoders of the prior art. Where the encoder receives a signal through the rate control line that the complexity threshold is about to be exceeded, the encoder may generate less events, to avoid exceeding that threshold for this next segment of the input data. In addition, as the controller typically triggers the addition of stuffing bits where the constraint threshold is exceeded, the processor may be implemented in a way that the processor need not account for, or deal with, the complexity constraint. Further, a decoder is described, capable of decoding an information sequence to an event sequence by accounting for a bounded ratio of events to information pieces in the information sequence, thereby allowing an efficient decoding of the information sequence for use in, for example, real-time use of the event sequence.

[0186] While this invention has been shown and described in connection with particular embodiments, it is apparent that certain changes and modifications in addition to those mentioned above may be made from the basic features of this invention. In addition, there may be many different types of computer software and hardware that may be utilized for practicing the invention, and the invention is not limited to the examples described above. Accordingly, the described embodiments are to be considered in all respects only as illustrative and not restrictive, and the scope of the invention is therefore indicated by the appended claims rather than the foregoing description. All changes that come within the scope and range of the claims are to be embraced within their scope.

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- [US6198848B1](#) [0008]
- [US6195465B1](#) [0009]
- [GB2330473A](#) [0010]

Non-patent literature cited in the description

- **MOFFAT et al.** Arithmetic Coding Revisited ACM Transactions on Information Systems, 1998, vol. 16, 3256-294 [\[0147\]](#)
- **RENE VAN DER VLEUTEN** Low Complexity Arithmetic Coding Implementation Document JVT-B033, JVT meeting in Geneva, 2002, [\[0184\]](#)
- **WINGER** Low-Complexity Arithmetic Coding Engine Document JVT-B036, JVT meeting in Geneva, 2002, [\[0184\]](#)

Patentkrav

1. Fremgangsmåde til aritmetisk afkodning til at konvertere en informationssekvens (735) omfattet af en række af bit til en binær hændelsessekvens (725) omfattet af en flerhed af binære hændelser, omfattende:

- 5 at modtage mindst en bit af informationssekvensen (735);
 at generere kontekstinformation for mindst en binær hændelse;
 at generere et sandsynlighedsestimat ($P(A)$) som reaktion på kontekstinformationen; og
- 10 at generere ved hensynstagen til et begrænset forhold af binære hændelser til bit, nul eller flere binære hændelser af den binære hændelsessekvens (725) som reaktion på den mindst ene modtagne bit og sandsynlighedsestimatet ($P(A)$);
 hvor hensynstagen til det begrænsede forhold af binære hændelser til bit inkluderer hensynstagen til et gennemsnitligt begrænset forhold af binære hændelser til bit;
- 15 hvor den aritmetiske afkodning anvender en aritmetisk afkoder (700), som inkluderer en kernemotor (715) og inkluderer at beregne (900) en underområdeværdi ($R(A)$) for en bestemt hændelsesværdi (A) under anvendelse af sandsynlighedsestimatet ($P(A)$);
- 20 hvor fremgangsmåden yderligere omfatter:
 at justere (910, 925, 930) mindst en af et områderegister (765) og et værdiregister (770) som reaktion på den mindst ene informationsbit og den beregnede underområdeværdi ($R(A)$),
 at udsende (915, 935) en hændelse (A , B) af hændelsessekvensen (725) som reaktion på en sammenligning (trin 905) mellem den beregnede underområdeværdi ($R(A)$) og værdiregisteret (770), og
- 25 kernemotoren (715) kalder en renormaliseringsrutine, renormaliseringsrutinen inkluderer hensynstagen til et gennemsnitligt forhold af hændelser til informationsbit, ved
- 30 at inkrementere (1005) et tælleregister (775) til at følge det gennemsnitlige forhold med en forudbestemt inkrementværdi ($1H$),

- hvor (trin 1000) en værdi (R) af områderegisteret (765) er $>$ en forudbestemt områdeværdi (4000H), og
- at dekrementere (1020) tælleregisteret (775) med en forudbestemt dekrementværdi (4H) hvor værdien (R) af områderegisteret (765) er
- 5 $<$ den forudbestemte områdeværdi (4000H);
- hvor at dekrementere (1020) tælleregisteret (775) inkluderer:
- at bestemme (1010) hvorvidt en værdi (C) af tælleregisteret (775) er $<$ nul,
- at fordoble (1015) værdiregisteret (770),
- 10 at tilføje (1015) en værdi (bit) af den udsendte hændelse til værdiregisteret (770),
- at dekrementere (1020) tælleregisteret (775) med den forudbestemte dekrementværdi (4H), når værdien (C) af tælleregisteret (775) ikke er $<$ nul, og
- 15 at fordoble (1025) områderegisteret (765),
- at fordoble (1015) værdiregisteret (770),
- at tilføje (1015) en værdi (bit) af den udsendte hændelse til værdiregisteret (770), og
- 20 at dekrementere (trin 1020) tælleregisteret (775) med den forudbestemte dekrementværdi (4H) når værdien (C) af tælleregisteret (775) er $<$ nul.

- 2.** Fremgangsmåde til aritmetisk kodning til at konvertere en binær hændelsessekvens (125) omfattet af en flerhed af binære hændelser til en
- 25 informationssekvens (135) omfattet af en række af bit, omfattende:

at modtage mindst en binær hændelse af den binære hændelsessekvens;

at generere kontekstinformation for den mindst ene binære hændelse;

at generere et sandsynlighedsestimat at den binære hændelse har en bestemt værdi som reaktion på kontekstinformationen; og

- at generere ved begrænsning af forholdet af binære hændelser til bit, nul eller flere bit af informationssekvensen som reaktion på den modtagne binære hændelse og sandsynlighedsestimatet, hvor begrænsningen af forholdet af binære hændelser til bit inkluderer begrænsning af det gennemsnitlige forhold af binære hændelser til bit;
- 5
- hvor den aritmetiske kodning anvender en aritmetisk koder, der inkluderer en kernemotor (115) og inkluderer at beregne en underområdeværdi for en bestemt hændelsesværdi under anvendelse af det tilsvarende sandsynlighedsestimat;
- 10
- hvor fremgangsmåden yderligere omfatter:
- at justere mindst en af et områderegister og et lavt register som reaktion på den modtagne hændelse og den beregnede underområdeværdi, og
- at kernemotoren (115) kalder renormaliseringsrutine,
- 15
- renormaliseringsrutinen inkluderer begrænsning af et gennemsnitligt forhold af hændelser til informationsstykker, ved
- at inkrementere (410) et tælleregister til at følge det gennemsnitlige forhold med en forudbestemt inkrementværdi hvor en hændelse behandles, og
- 20
- at dekrementere tælleregisteret med en forudbestemt dekrementværdi hvor et informationsstykke genereres;
- hvor at dekrementere tælleregisteret inkluderer:
- at bestemme (430) hvorvidt en værdi af tælleregisteret er $< \text{nul}$, og
- at dekrementere (435) tælleregisteret med den forudbestemte dekrementværdi når værdien af tælleregisteret ikke er $< \text{nul}$, og
- 25
- at fordoble (440) værdien af områderegisteret, og
- at dekrementere (435) tælleregisteret med den forudbestemte dekrementværdi når værdien af tælleregisteret er $< \text{nul}$.

DRAWINGS

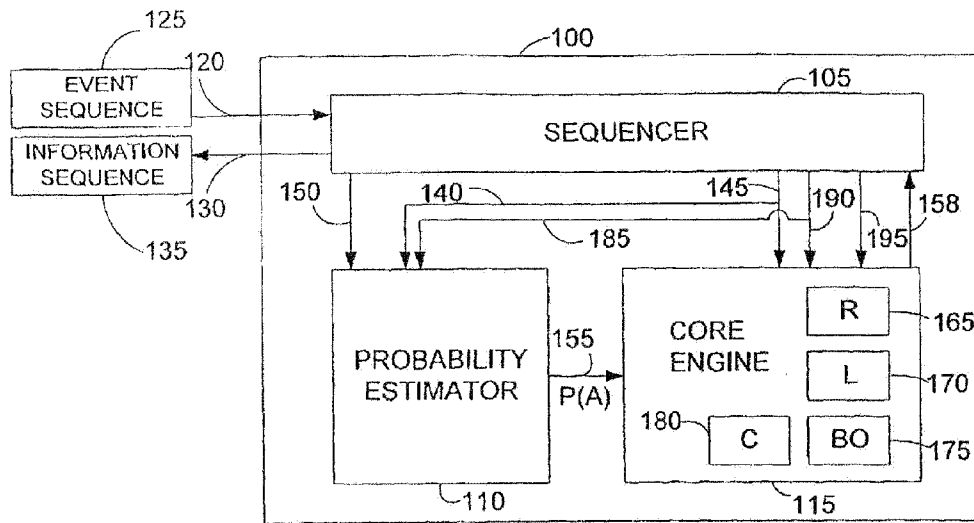


FIGURE 1

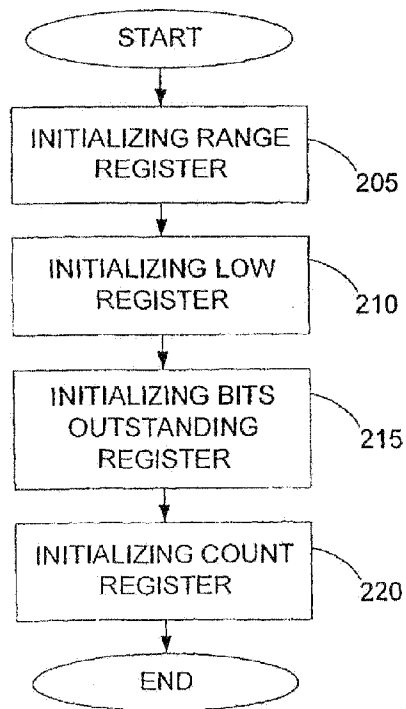


FIGURE 2

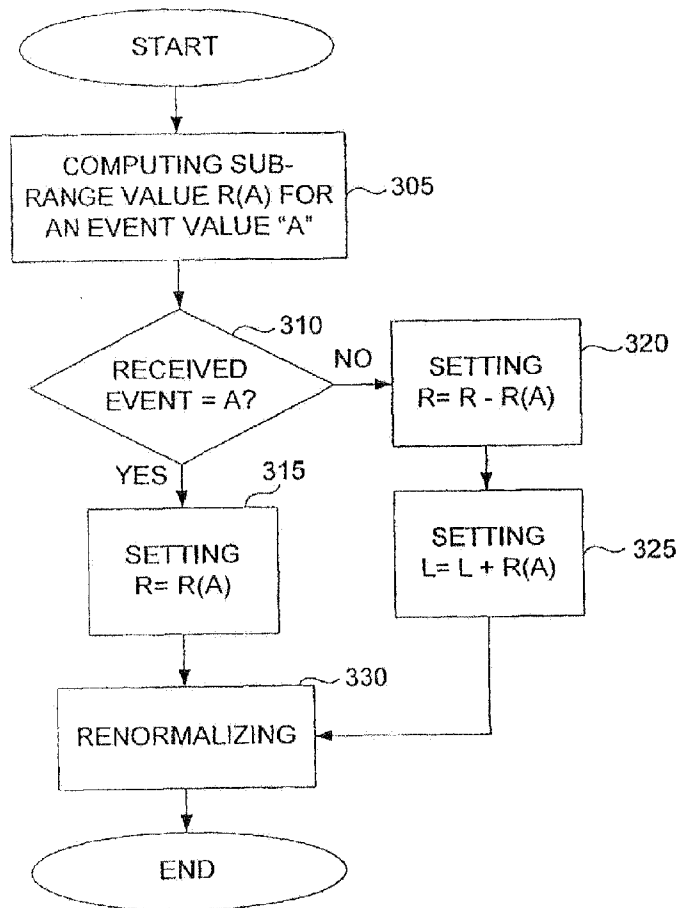


FIGURE 3

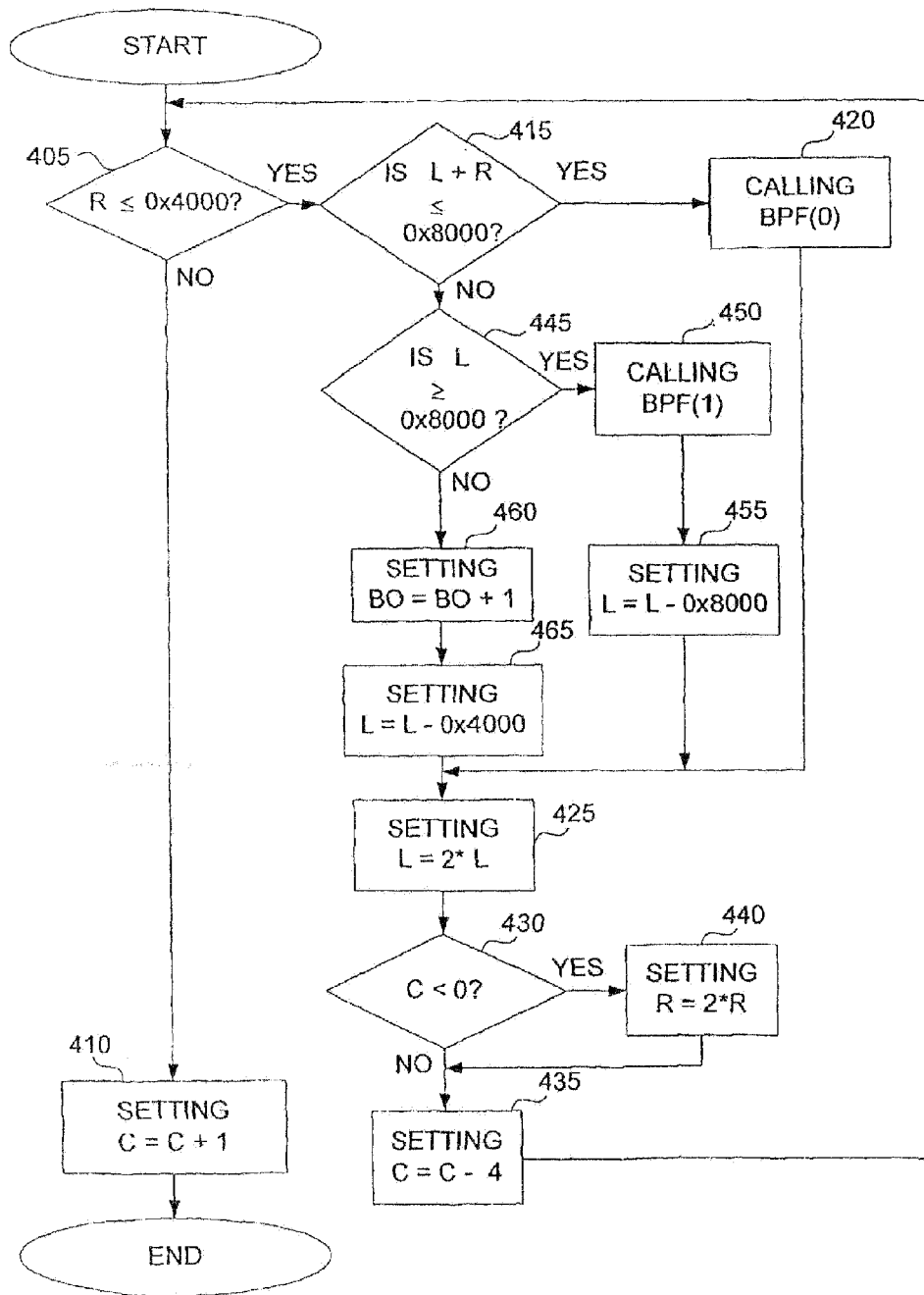


FIGURE 4

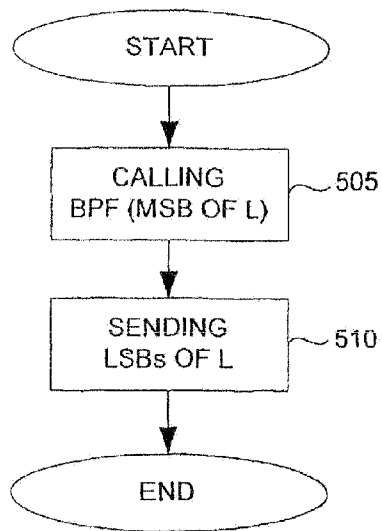


FIGURE 5

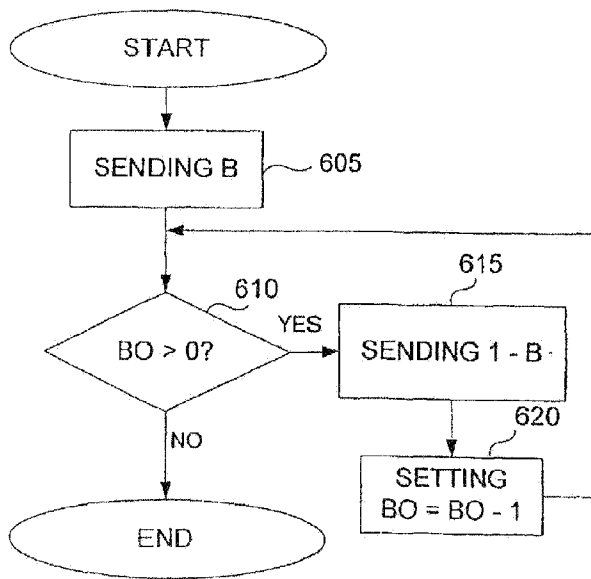


FIGURE 6

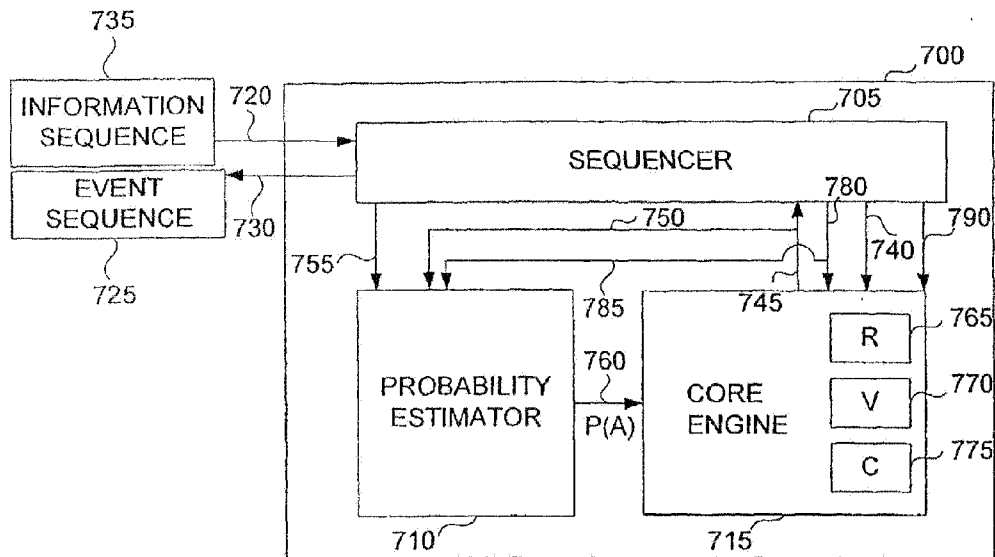


FIGURE 7

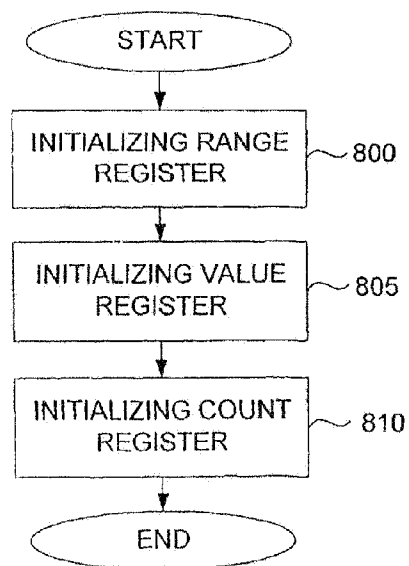


FIGURE 8

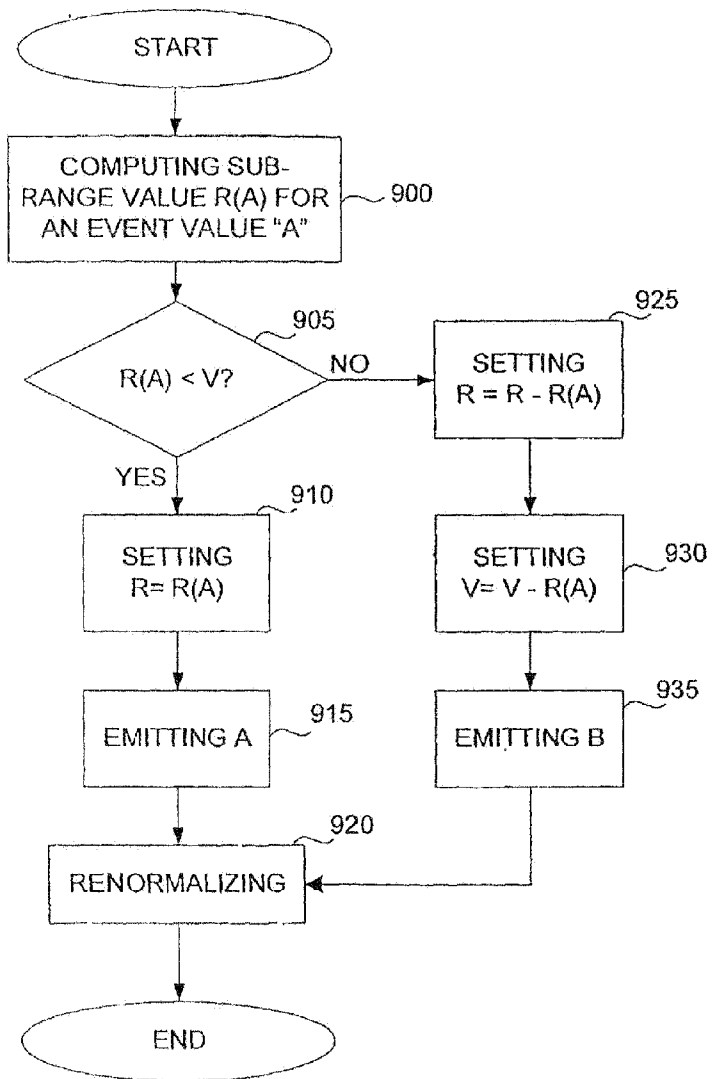


FIGURE 9

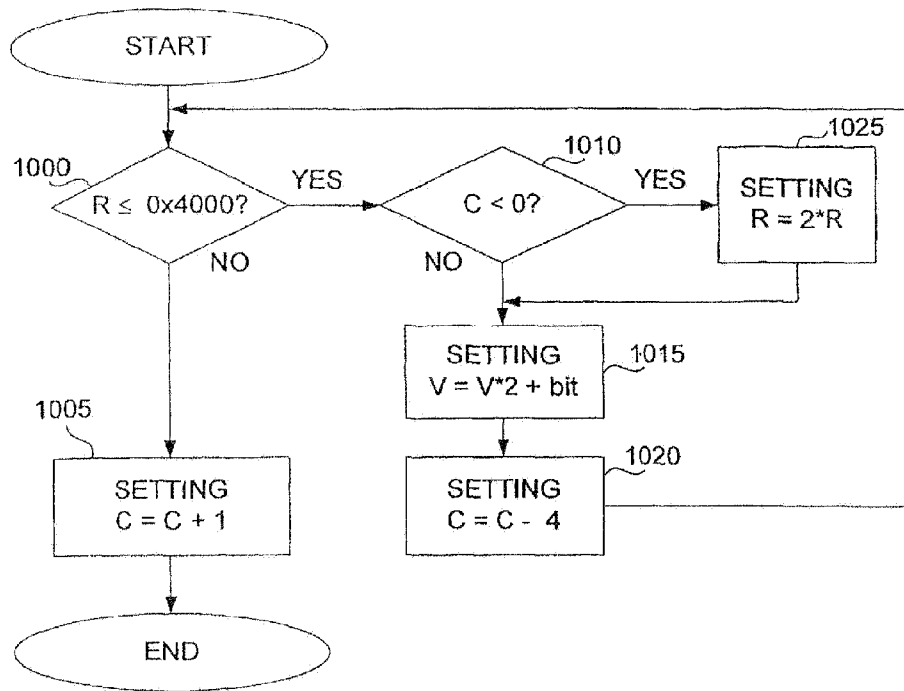


FIGURE 10

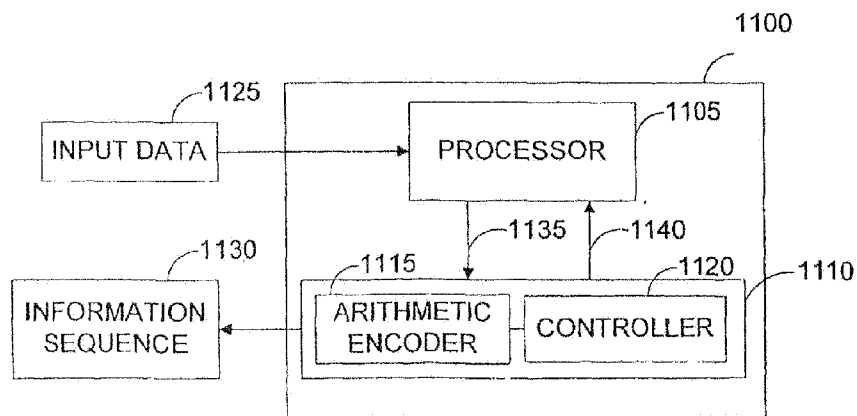


FIGURE 11