

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

(19) World Intellectual Property Organization

International Bureau



(43) International Publication Date
25 June 2015 (25.06.2015)



(10) International Publication Number

WO 2015/092483 A1

(51) International Patent Classification:

G10L 19/038 (2013.01)

(21) International Application Number:

PCT/IB2013/061034

(22) International Filing Date:

17 December 2013 (17.12.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: NOKIA TECHNOLOGIES OY [FI/FI];
Karaportti 3, FI-02610 Espoo (FI).

(72) Inventors: VASILACHE, Adriana; Koiramaenkatu 27,
FI-33580 Tampere (FI). RAMO, Anssi, Sakari; Veholehdenkatu 9, FI-33720 Tampere (FI). LAAKSONEN, Lasse, Juhani; Tiesenkuja 1 B 4, FI-37120 Tampere (FI).

(74) Agent: SMITH, Gary, John; Page White & Farrer, Bedford House, John Street, London Greater London WC1N 2BF (GB).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,

DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

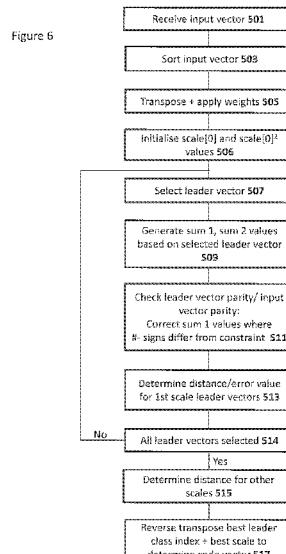
Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

— with international search report (Art. 21(3))

(54) Title: AUDIO SIGNAL ENCODER



(57) Abstract: An apparatus comprising: a vector generator configured to generate at least one vector of parameters defining at least one audio signal; a lattice vector quantizer configured to sort the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters; the lattice vector quantizer configured to select from a list of leader classes at least one potential code vector; the lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters; the lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance; the lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector.

WO 2015/092483 A1

Audio Signal Encoder

Field

5 The present application relates to a multichannel or stereo audio signal encoder, and in particular, but not exclusively to a multichannel or stereo audio signal encoder for use in portable apparatus.

Background

10

Audio signals, like speech or music, are encoded for example to enable efficient transmission or storage of the audio signals.

15 Audio encoders and decoders (also known as codecs) are used to represent audio based signals, such as music and ambient sounds (which in speech coding terms can be called background noise). These types of coders typically do not utilise a speech model for the coding process, rather they use processes for representing all types of audio signals, including speech. Speech encoders and decoders (codecs) can be considered to be audio codecs which are 20 optimised for speech signals, and can operate at either a fixed or variable bit rate.

25 Audio encoders and decoder are often designed as low complexity source coders. In other words able to perform encoding and decoding of audio signals without requiring highly complex processing.

30 An example of which is transform coding. For music signal audio encoding transform coding generally performs better than Algebraic Code Excited Linear Prediction (ACELP) technology which is better suited and directed for speech signals. Transform coding is performed by coding transform coefficients vector sub-band wise. In other words an audio signal is divided into sub-bands for

which a parameter is determined and the parameters represent sub-vectors which are vector or lattice quantised.

Summary

5

According to a first aspect there is provided a method comprising: generating at least one vector of parameters defining at least one audio signal; sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters; selecting from a list of leader classes at least one potential code vector; determining a distance between the at least one potential code vector and the at least one ordered vector of parameters; determining at least one leader class associated with a potential code vector which generates the smallest associated distance; transposing the at least one leader class to generate an output lattice quantized codevector.

The method may further comprise: selecting at least one scale factor; determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may further comprise determining a distance based on the at least one scale factor; determining at least one leader class associated with a potential code vector which generates the smallest associated distance may further comprise determining an output scale factor associated with a potential code vector and scale factor which generates the smallest associated distance; transposing the at least one leader class to generate an output lattice quantized codevector may comprise applying the output scale factor to the output lattice quantized codevector.

Generating a first vector of parameters defining at least one audio signal may comprise: dividing the at least one audio signal into time frames; determining a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

Sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters may comprise: determining weights for a weighted distance determination; sorting the weights based on the ordering of the at least

5 one vector absolute tuples to generate a sorted weight vector; applying the sorted weight vector to the at least one ordered vector of parameters.

Determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may comprise: generating a first and

10 second distance components based on a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered vector of parameters; correcting the sign of the first distance component where the potential code vector fails a parity constraint test; determining a distance between the at least 15 one potential code vector and the at least one ordered vector of parameters based on the first and second distance components.

Determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may further comprise determining a

20 distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components and further scale values.

Determining at least one leader class associated with a potential code vector

25 which generates the smallest associated distance may comprise: determining the index of the potential code vector which generates the smallest associated distance; determining the scale value of the potential code vector which generates the smallest associated distance by applying scale values to only the potential code vector which generates the smallest associated distance.

Transposing the at least one leader class to generate an output lattice quantized codevector may comprise sorting the at least one leader class by the reverse ordering of at least one vector absolute tuples.

- 5 According to a second aspect there is provided an apparatus comprising: means for generating at least one vector of parameters defining at least one audio signal; means for sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters; means for selecting from a list of
- 10 10 leader classes at least one potential code vector; means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters; means for determining at least one leader class associated with a potential code vector which generates the smallest associated distance; means for transposing the at least one leader class to generate an
- 15 15 output lattice quantized codevector.

The apparatus may further comprise: means for selecting at least one scale factor; wherein the means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may further comprise means for determining a distance based on the at least one scale factor; the means for determining at least one leader class associated with a potential code vector which generates the smallest associated distance may further comprise means for determining an output scale factor associated with a potential code vector and scale factor which generates the smallest associated distance; the means for transposing the at least one leader class to generate an output lattice quantized codevector may comprise means for applying the output scale factor to the output lattice quantized codevector.

The means for generating a first vector of parameters defining at least one audio signal may comprise: means for dividing the at least one audio signal into time frames; means for determining a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

The means for sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters may comprise: means for determining

5 weights for a weighted distance determination; means for sorting the weights based on the ordering of the at least one vector absolute tuples to generate a sorted weight vector; means for applying the sorted weight vector to the at least one ordered vector of parameters.

10 The means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may comprise: means for generating a first and second distance components based on a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered

15 vector of parameters; means for correcting the sign of the first distance component where the potential code vector fails a parity constraint test; means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components.

20

The means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may further comprise means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and

25 second distance components and further scale values.

The means for determining at least one leader class associated with a potential code vector which generates the smallest associated distance may comprise: means for determining the index of the potential code vector which generates

30 the smallest associated distance; means for determining the scale value of the potential code vector which generates the smallest associated distance by

applying scale values to only the potential code vector which generates the smallest associated distance.

The means for transposing the at least one leader class to generate an output

5 lattice quantized codevector may comprise means for sorting the at least one leader class by the reverse ordering of at least one vector absolute tuples.

According to a third aspect there is provided an apparatus comprising at least one processor and at least one memory including computer program code, the

10 at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to: generate at least one vector of parameters defining at least one audio signal; sort the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters;

15 select from a list of leader classes at least one potential code vector; determine a distance between the at least one potential code vector and the at least one ordered vector of parameters; determine at least one leader class associated with a potential code vector which generates the smallest associated distance; transpose the at least one leader class to generate an output lattice quantized

20 codevector.

The apparatus may be further caused to: select at least one scale factor; wherein determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may further cause the

25 apparatus to determine a distance based on the at least one scale factor; wherein determining at least one leader class associated with a potential code vector which generates the smallest associated distance may further cause the apparatus to determine an output scale factor associated with a potential code vector and scale factor which generates the smallest associated distance;

30 wherein transposing the at least one leader class to generate an output lattice quantized codevector may cause the apparatus to apply the output scale factor to the output lattice quantized codevector.

Generating a first vector of parameters defining at least one audio signal may cause the apparatus to: divide the at least one audio signal into time frames; determine a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

Sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters may cause the apparatus to: determine weights for a weighted distance determination; sort the weights based on the ordering of the at least one vector absolute tuples to generate a sorted weight vector; apply the sorted weight vector to the at least one ordered vector of parameters.

Determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may cause the apparatus to: generate a first and second distance components based on a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered vector of parameters; correct the sign of the first distance component where the potential code vector fails a parity constraint test; determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components.

Determining a distance between the at least one potential code vector and the at least one ordered vector of parameters may further cause the apparatus to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components and further scale values.

Determining at least one leader class associated with a potential code vector which generates the smallest associated distance may cause the apparatus to: determine the index of the potential code vector which generates the smallest

associated distance; determine the scale value of the potential code vector which generates the smallest associated distance by applying scale values to only the potential code vector which generates the smallest associated distance.

5

Transposing the at least one leader class to generate an output lattice quantized codevector may cause the apparatus to sort the at least one leader class by the reverse ordering of at least one vector absolute tuples.

10 According to a fourth aspect there is provided an apparatus comprising: a vector generator configured to generate at least one vector of parameters defining at least one audio signal; a lattice vector quantizer configured to sort the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters; the lattice vector quantizer configured to select from a list of leader classes at least one potential code vector; the lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters; the lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance; the lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector.

15

20 The apparatus may further comprise: the lattice vector quantizer configured to select at least one scale factor; wherein the lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters further may be configured to determine a distance based on the at least one scale factor; the lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance may be further configured to determine an output scale factor associated with a potential code vector and scale factor which generates the smallest associated

25

30

distance; the lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector may be further configured to apply the output scale factor to the output lattice quantized codevector.

5

The apparatus may further comprise a parameter determiner configured to divide the at least one audio signal into time frames; and determine a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

10

Wherein the lattice vector quantizer configured to sort the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters may be further configured to: determine weights for a weighted distance determination;

15 sort the weights based on the ordering of the at least one vector absolute tuples to generate a sorted weight vector; apply the sorted weight vector to the at least one ordered vector of parameters.

The lattice vector quantizer configured to determine a distance between the at 20 least one potential code vector and the at least one ordered vector of parameters may be further configured to: generate a first and second distance components based on a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered vector of parameters; correct the sign 25 of the first distance component where the potential code vector fails a parity constraint test; determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components.

30 The lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters further may be configured to determine a distance between the at

least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components and further scale values.

- 5 The lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance may be further configured to: determine the index of the potential code vector which generates the smallest associated distance; determine the scale value of the potential code vector which generates the smallest associated
- 10 distance by applying scale values to only the potential code vector which generates the smallest associated distance.

The lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector may be configured to

- 15 sort the at least one leader class by the reverse ordering of at least one vector absolute tuples.

A computer program product may cause an apparatus to perform the method as described herein.

- 20 An electronic device may comprise apparatus as described herein.

A chipset may comprise apparatus as described herein.

Brief Description of Drawings

- 25 For better understanding of the present invention, reference will now be made by way of example to the accompanying drawings in which:

Figure 1 shows schematically an electronic device employing some embodiments;

- 30 Figure 2 shows schematically an audio codec system according to some embodiments;

Figure 3 shows schematically an encoder as shown in Figure 2 according to some embodiments;

Figure 4 shows a flow diagram illustrating the operation of the encoder shown in Figure 3 according to some embodiments;

5 Figure 5 shows schematically a lattice vector quantizer as shown in Figure 3 according to some embodiments; and

Figure 6 shows a flow diagram illustrating the operation of the lattice vector quantizer shown in Figure 5 according to some embodiments;.

10 Description of Some Embodiments of the Application

The following describes in more detail possible stereo and multichannel speech and audio codecs, including layered or scalable variable rate speech and audio codecs.

15

There can be a problem with current transform coding approaches in that the use of compression efficient lattices can improve significantly the quantisation. However they manage to produce such improvements at the cost of significant codec complexity.

20

The concept as discussed in detail by the embodiments herein propose an approach allowing for significant encoding complexity reduction by evaluating the quantization distortion in a transposed vector space.

25 In this regard reference is first made to Figure 1 which shows a schematic block diagram of an exemplary electronic device or apparatus 10, which may incorporate a codec according to an embodiment of the application.

30 The apparatus 10 may for example be a mobile terminal or user equipment of a wireless communication system. In other embodiments the apparatus 10 may be an audio-video device such as video camera, a Television (TV) receiver, audio recorder or audio player such as a mp3 recorder/player, a media recorder

(also known as a mp4 recorder/player), or any computer suitable for the processing of audio signals.

5 The electronic device or apparatus 10 in some embodiments comprises a microphone 11, which is linked via an analogue-to-digital converter (ADC) 14 to a processor 21. The processor 21 is further linked via a digital-to-analogue (DAC) converter 32 to loudspeakers 33. The processor 21 is further linked to a transceiver (RX/TX) 13, to a user interface (UI) 15 and to a memory 22.

10 The processor 21 can in some embodiments be configured to execute various program codes. The implemented program codes in some embodiments comprise an audio encoding or decoding code as described herein. The implemented program codes 23 can in some embodiments be stored for example in the memory 22 for retrieval by the processor 21 whenever needed.

15 The memory 22 could further provide a section 24 for storing data, for example data that has been encoded in accordance with the application.

The encoding and decoding code in embodiments can be implemented at least partially in hardware and/or firmware.

20 The user interface (UI) 15 enables a user to input commands to the electronic device 10, for example via a keypad, and/or to obtain information from the electronic device 10, for example via a display. In some embodiments a touch screen may provide both input and output functions for the user interface. The apparatus 10 in some embodiments comprises a transceiver (RX/TX) 13 suitable for enabling communication with other apparatus, for example via a wireless communication network.

25 The transceiver 13 can communicate with further devices by any suitable known communications protocol, for example in some embodiments the transceiver 13 or transceiver means can use a suitable universal mobile telecommunications system (UMTS) protocol, a wireless local area network (WLAN) protocol such

as for example IEEE 802.X, a suitable short-range radio frequency communication protocol such as Bluetooth, or infrared data communication pathway (IRDA).

5 It is to be understood again that the structure of the apparatus 10 could be supplemented and varied in many ways.

A user of the apparatus 10 for example can use the microphone 11 for inputting speech or other audio signals that are to be transmitted to some other 10 apparatus or that are to be stored in the data section 24 of the memory 22. A corresponding application in some embodiments can be activated to this end by the user via the user interface 15. This application in these embodiments can be performed by the processor 21, causes the processor 21 to execute the encoding code stored in the memory 22. Although in the following examples the 15 microphone 11 is configured to generate the audio signals for inputting it would be understood that the input audio signals can be received from any suitable input such as from the memory 22 and specifically within the stored data 24 section of the memory 22. In some embodiments the input audio signal or at least one audio signal can be received via the transceiver 13. For example the 20 transceiver 13 can be configured to receive audio signals generated by microphones external to the apparatus 10, for example a Bluetooth device coupled to the apparatus via the transceiver 13.

The analogue-to-digital converter (ADC) 14 in some embodiments converts the 25 input analogue audio signal into a digital audio signal and provides the digital audio signal to the processor 21. In some embodiments the microphone 11 can comprise an integrated microphone and ADC function and provide digital audio signals directly to the processor for processing.

30 The processor 21 in such embodiments then processes the digital audio signal in the same way as described with reference to the system shown in Figure 2,

and specifically the encoder shown in Figures 3, and details of the encoder shown in Figures 5.

The resulting bit stream can in some embodiments be provided to the

5 transceiver 13 for transmission to another apparatus. Alternatively, the coded audio data in some embodiments can be stored in the data section 24 of the memory 22, for instance for a later transmission or for a later presentation by the same apparatus 10.

10 The apparatus 10 in some embodiments can also receive a bit stream with correspondingly encoded data from another apparatus via the transceiver 13. In this example, the processor 21 may execute the decoding program code stored in the memory 22. The processor 21 in such embodiments decodes the received data, and provides the decoded data to a digital-to-analogue converter 32. The digital-to-analogue converter 32 converts the digital decoded data into 15 analogue audio data and can in some embodiments output the analogue audio via the loudspeakers 33. Execution of the decoding program code in some embodiments can be triggered as well by an application called by the user via the user interface 15.

20

The received encoded data in some embodiment can also be stored instead of an immediate presentation via the loudspeakers 33 in the data section 24 of the memory 22, for instance for later decoding and presentation or decoding and forwarding to still another apparatus.

25

It would be appreciated that the schematic structures described in Figures 3 and 5 and the method steps shown in Figures 4 and 6 represent only a part of the operation of an audio codec and specifically part of an audio encoder apparatus or method as exemplarily shown implemented in the apparatus 30 shown in Figure 1.

The general operation of audio codecs as employed by embodiments is shown in Figure 2. General audio coding/decoding systems comprise both an encoder and a decoder, as illustrated schematically in Figure 2. However, it would be understood that some embodiments can implement one of either the encoder or

5 decoder, or both the encoder and decoder. Illustrated by Figure 2 is a system 102 with an encoder 104, a storage or media channel 106 and a decoder 108. It would be understood that as described above some embodiments can comprise or implement one of the encoder 104 or both the encoder 104 and decoder 108.

10 The encoder 104 compresses an input audio signal 110 producing a bit stream 112, which in some embodiments can be stored or transmitted through a media channel 106. The encoder 104 can in some embodiments comprise a multi-channel encoder that encodes two or more audio signals.

15 The bit stream 112 can be received within the decoder 108. The decoder 108 decompresses the bit stream 112 and produces an output audio signal 114. The decoder 108 can comprise a transform decoder as part of the overall decoding operation. The decoder 108 can also comprise a multi-channel decoder that decodes two or more audio signals. The bit rate of the bit stream 112 and the 20 quality of the output audio signal 114 in relation to the input signal 110 are the main features which define the performance of the coding system 102.

Figure 3 shows schematically the encoder 104 according to some embodiments.

25 Figure 4 shows schematically in a flow diagram the operation of the encoder 104 according to some embodiments.

The concept for the embodiments as described herein is to determine and apply 30 encoding to audio signals to produce efficient high quality and low bit rate real life coding. To that respect with respect to Figure 3 an example encoder 104 is shown according to some embodiments. Furthermore with respect to Figure 4 the operation of the encoder 104 is shown in further detail. In the following

examples the encoder is configured to generate frequency domain parameters representing the audio signal and encode the generated frequency domain parameters using a suitable vector lattice quantization, however it would be understood that in some embodiments the parameters used in the lattice quantization as described herein can be any suitable parameters defining or representing the audio signals or other type of signals (for example image or, video).

5 The encoder 104 in some embodiments comprises a frame sectioner 201 or suitable means for sectioning the audio signal. The frame sectioner 201 is configured to receive the audio signals (for example a mono, left and right stereo or any multichannel audio representation) input audio signal and section or segment the audio signal data into sections or frames suitable for frequency or other domain transformation. The frame sectioner 201 in some embodiments 10 can further be configured to window these frames or sections of audio signal data according to any suitable windowing function. For example the frame sectioner 201 can be configured in some embodiments to generate frames of 20ms which overlap preceding and succeeding frames by 10ms each.

15

20 The operation of generating audio frames is shown in Figure 4 by step 501.

In some embodiments the audio frames can be passed to a parameter determiner 203.

25 In some embodiments the encoder comprises a parameter determiner 203 of suitable means for determining at least one parameter representing the input audio signal(s) or input audio signal frames. In the following examples the parameter is a line spectral frequency (LSF) parameter however it would be understood that in some embodiments any suitable parameter can be 30 determined.

For example in some embodiments the parameter determiner comprises a transformer 203 or suitable means for transforming. The transformer 203 in some embodiments is configured to generate frequency domain (or other suitable domain) parameter representations of these audio signals. These 5 frequency domain parameter representations can in some embodiments be passed to the parameter encoder 205.

In some embodiments the transformer 203 can be configured to perform any suitable time to frequency domain transformation on the audio signal data. For 10 example the time to frequency domain transformation can be a discrete Fourier transform (DFT), Fast Fourier transform (FFT), modified discrete cosine transform (MDCT). In the following examples a Fast Fourier Transform (FFT) is used.

15 Furthermore the transformer can further be configured to generate separate frequency band domain parameter representations (sub-band parameter representations) of each input channel audio signal data. These bands can be arranged in any suitable manner. For example these bands can be linearly spaced, or be perceptual or psychoacoustically allocated. The parameters 20 generated can be any suitable parameter.

The operation of determining or generating parameter representations is shown in Figure 4 by step 503.

25 In some embodiments the representations, such as LSF parameters, are passed to a parameter encoder 205.

In some embodiments the encoder 104 can comprise a parameter encoder 205. The parameter encoder 205 can be configured to receive the parameter 30 representations of the audio signal input, for example the determined LSF parameters. The parameter encoder 205 can furthermore in some embodiments be configured to use each of the LSF parameter values as a sub-vector and

combine each sub-vector to create a vector to input into a vector quantizer. In other words the apparatus can comprise a vector generator configured to generate a first vector of parameters (or tuples of a first vector representing the parameters) defining at least one audio signal.

5

The output of the vector quantizer is in some embodiments the encoder and therefore the vector quantized audio signals output are the 'encoded' or parameter encoded representations of the audio signal.

10 The operation of encoding or vector quantizing the parameters is shown in Figure 4 by step 505

In some embodiments the parameter encoder 205 comprises a vector generator 451. The vector generator 451 is configured to receive the LSF parameters and 15 generate a N dimensional vector from these values.

The operation of generating vectors from the input parameters is shown in Figure 4 by sub-step 551.

20 The generated vectors can in some embodiments be passed to the lattice vector quantizer 453.

In some embodiments the parameter encoder 205 comprises a lattice vector quantizer 453. The lattice vector quantizer 453 receives the input vector 25 generated from the LSF parameters and generates a nearest neighbour or NN output which occurs within a defined lattice and thus can be decoded using a similar lattice at the decoder.

30 The operation of Lattice quantizing the vector is shown in Figure 4 by sub-step 553.

The encoded signal can be output.

The operation of outputting the encoded signal is shown in Figure 4 by step 507. This for the example can be an operation of outputting the quantized lattice vector as shown in Figure 4 by sub-step 557.

5

With reference to Figure 5 there is shown an example lattice vector quantizer 453 according to some embodiments. The lattice quantizer 453 can in some embodiments be defined by respective program code 23 of a computer program that is stored on a tangible storage medium memory 22.

10

Before introducing the concepts and embodiments with respect to the invention we shall initially discuss conventional lattice vector quantization. In some lattice quantizers an initial generating or determining a set of potential basis code vectors, wherein each determined potential basis code vector of this set of potential basis code vectors is associated with a potential basis code vector of a different set of basis code vectors is performed.

Each set of potential basis code vectors comprises at least one basis code vector. Since each set of basis code vectors is associated with at least one scale representative of a plurality of scale representatives, a code vector can be determined based on a basis code vector of a set of potential basis code vectors and a scale representative of the at least one scale representative associated with the set of potential basis code vectors. In other words the code vector may be represented based on a basis code vector scaled by the respective scale representative. For instance, the scale representative may represent a scale value, wherein a code vector may be determined based on a multiplication of a basis code vector and the respective scale value. Furthermore in some embodiments the codebook is obtained by applying a (signed) permutation of the basis vector.

20
25
30

For instance, at least one set of basis code vectors is associated with at least two scale representatives.

Accordingly, as an example, a codebook may comprise a set of code vectors comprising code vectors based on the plurality of sets of basis code vectors and based on the respective at least one scale value associated with a respective set of basis code vectors of the plurality of basis code vectors. This set of code vectors may comprise, for each basis code vector of each set of basis code vectors and for each of the at least one scale representative associated with a respective set of basis code vectors, a code vector based on the respective basis code vector scaled by the respective scale representative.

10

For instance, said sets of basis code vectors may represent leader classes, wherein each leader class comprises a different leader vector and permutations of said leader vector. Thus, said leader vector and the permutations of said leader vector may represent the basis code vectors of the respective set of basis code vectors.

The plurality of sets of basis code vectors may represent a subset of a second plurality of sets of basis code vectors. For instance, under the assumption that each set of basis code vector represents a leader class, the plurality of leader classes may represent a subset of a second plurality of leader classes. Thus, the plurality of leader classes may be considered as a truncated plurality of leader classes with respect to the second plurality of leader classes.

For instance, the respective potential basis code vector may be determined by determining the basis code vector of the at least one basis code vector of the respective set of basis code vector which is nearest to the input vector to be encoded. Any kind of suitable criterion may be used for finding the nearest basis code vector with respect to the input vector to be encoded.

30 As an example, a potential basis code vector may be determined based on a nearest basis code vector with respect to the absolute valued input vector and based on information of signs of the values of the input vector, wherein this

information may comprise the sign of a respective position of respective values in the input vector and is used to assign signs to values of the determined potential basis code vector. Furthermore, as an example, the basis code vector which is nearest to the absolute valued input vector may be determined,

5 wherein the absolute valued input vector comprises absolute values corresponding to the values of the input vector, wherein the potential basis code vector represents the determined nearest basis code vector, wherein the signs of the values of the potential basis code vector correspond to the signs of the values of the input vector at the same position in the vector, wherein this may

10 hold if the parity of the basis code vectors of the set of basis code vectors is 0. As another example, if the parity of the basis code vectors of the set of basis code vectors is -1, the signs of the values of the potential basis code vector may be assigned corresponding to the signs of the values of the input vector at the same position in the vector, respectively, and if there are not an odd number of

15 negative components, the value in the potential basis code vector having the lowest non-null absolute value may change its sign. Or, as another example, if the parity of the basis code vectors of the set of basis code vectors is +1, the signs of the values of the potential basis code vector may be assigned corresponding to the signs of the values of the input vector at the same position

20 in the vector, respectively, and if there are not an even number of negative components, the value in the potential basis code vector having the lowest non-null absolute value may change its sign.

The code vector for encoding the input vector is then conventionally determined

25 based on the set of determined potential code vectors, wherein said set of determined potential code vectors defines a subset of code vectors, said subset of code vectors comprising, for each determined potential basis code vector and each scale representative associated with the set of basis code vectors of the respective potential basis code vector, a code vector based on the respective potential basis code vector scaled by the respective scale representative.

Accordingly, the search for the code vector for encoding the input vector has been performed in the subset of code vectors defined by the determined potential code vectors and defined by the respective at least one scale representative associated with the set of basis code vectors of the respective 5 determined potential code vector. Since this subset of code vectors may represent a subset of code vectors associated with the codebook, the number of code vectors of this subset of code vectors may be less than the number of code vectors of the set of code vectors.

10 As an example, each scale representative of the plurality of scale representatives may be associated with at least one set of code vectors, wherein each set of code vectors of said at least one set of code vectors associated with a respective scale representative is associated with a set of basis code vectors of the plurality of sets of basis code vectors such that each 15 set of code vectors of said at least one set of code vectors associated with a respective scale representative comprises code vectors obtained by scaling the basis vectors of the associated respective set of basis vectors with the respective scale representative.

20 Accordingly, the code vectors of the at least one set of basis code vectors associated with a respective scale representative of the plurality of scale representatives can be determined based on scaling the basis code vectors of each set of basis code vectors associated with the scale representative with this scale representative.

25

For instance, in case said sets of basis code vectors represent leader classes, the at least one set of basis code vectors associated with a respective scale representative may be considered as a union of leader classes. It would be understood that usually the union of leader classes is independent of the scale.

30 Thus, the codebook may comprise at last one union of leader classes, wherein each union of leader class is associated with one of at least one scale representatives and with at least one set of basis code vectors of the plurality of

basis code vectors. As an example, the at least one scale representative may represent the plurality of scale representatives which may comprise at least two scale representatives.

- 5 Thus for example b_x , with $x \in \{0, 1, \dots, X-1\}$, represents a set of basis code vectors of the plurality of sets of basis code vectors, wherein X represents the number of sets of the plurality of sets of basis code vectors. Each set of basis code vectors is associated or comprises at least one basis code vector $b_{x,y}$, wherein B_x represents the number of basis code vectors of a respective set of basis code vectors b_x , i.e. $y \in \{0, 1, \dots, B_x-1\}$ holds. For instance, the number B_x of basis code vectors of a set of basis code vectors may be different for different sets of basis code vectors and/or it may be the same for at least two sets of basis code vectors.
- 10
- 15 In other words a leader vector is just one vector. Together with all the signed permutations of the leader vector then this set forms the leader vector's leader class (or as described herein the basis code vectors). When putting together several leader classes, a union of leader classes is formed. Then to this union/unions one or more scales can be attached.
- 20 Thus for example it may be possible to determine a code vector $c_{x,z,y}$ based on basis code vector $b_{x,y}$ and based on a scale representative s_z , wherein index z represents the index of the respective scale representative of the plurality of scale representatives $s_0 \dots s_{S-1}$, i.e. $z \in \{0, 1, \dots, S-1\}$ holds.
- 25 For instance, in case the values $b_{x,y,t}$ of the basis code vectors $b_{x,y} = [b_{x,y,0}, b_{x,y,1}, \dots, b_{x,y,n-1}]$ represent absolute values, wherein $t \in \{0, 1, \dots, n-1\}$ holds and n represents the length of the respective basis code vector $b_{x,y}$, and if the absolute valued input vector is used for determining the potential code vector of a respective set of basis code vectors, the sign of each value $b_{x,y,t}$ at the $(t+1)$ th position of the determined nearest basis code vector $b_{x,y}$ may be assigned based on the sign of the respective value i_t at the $(t+1)$ th position of the input
- 30

vector i , before determining a code vector $c_{x,z,y}$ based on basis code vector $b_{x,y}$ and based on a scale representative s_z is performed.

As an example, if $i=[i_0, i_1, \dots, i_{n-1}]$ represents the input vector, the absolute valued

5 input vector may be represented by $||i_0||, ||i_1||, \dots, ||i_{n-1}||$. For instance, the sign of each value $b_{x,y,t}$ at the $(t+1)$ th position of the determined nearest basis code vector $b_{x,y}$ may be assigned to the sign of the respective value i_t at the $(t+1)$ th position of the input vector, respectively, wherein this may hold if the parity of the basis code vectors $b_{x,y}$ of the set of basis code vectors b_x is 0. As another
10 example, if the parity of the basis code vectors $b_{x,y}$ of the set of basis code vectors b_x is -1, the signs of the values $b_{x,y,t}$ of the potential basis code vector may be assigned corresponding to the signs of the values of the input vector at the same position in the vector, respectively, and if there are not an odd number of negative components, the value $b_{x,y,t}$ in the potential basis code vector having
15 the lowest non-null absolute value may change its sign. Or, as another example, if the parity of the basis code vectors $b_{x,y}$ of the set of basis code vectors b_x is +1, the signs of the values $b_{x,y,t}$ of the potential basis code vector may be assigned corresponding to the signs of the values of the input vector at the same position in the vector, respectively, and if there are not an even
20 number of negative components, the value $b_{x,y,t}$ in the potential basis code vector having the lowest non-null absolute value may change its sign.

As a non-limiting example, a code vector $c_{x,z,y}$ may be determined by $c_{x,z,y} = [b_{x,y,0} \cdot s_z, b_{x,y,1} \cdot s_z, \dots, b_{x,y,n-1} \cdot s_z]$.

25

Each of the scale representatives s_z , wherein $z \in \{0, 1, \dots, S-1\}$ holds, is associated with at least one set of basis code vectors. For instance, as a non-limiting example this respective at least one set of basis code vectors may be represented by the set of basis code vectors b_x , with $x \in \{0, 1, \dots, n_z - 1\}$, wherein
30 n_z may represent the number of sets of basis code vectors associated with the respective scale representative s_z , wherein $0 < n_z < X$ holds. Based on this linkage between a respective scale representative s_z and the associated at least one

set of basis code vectors b_x , with $x \in \{0, 1, \dots, n_x-1\}$, the associated at least one set of code vectors $c_{x,z,y}$, with $x \in \{0, 1, \dots, n_x-1\}$ and $y \in \{0, 1, \dots, B_x-1\}$ and $z \in \{0, 1, \dots, S-1\}$, can be determined.

- 5 Thus, as an example, a codebook structure of the above mentioned codebook may be defined by the plurality of scale representatives s_z , the plurality of sets of basis code vectors b_x , and the linkage between each scale representative with the associated at least one set of basis code vectors.
- 10 Since at least one set of basis code vectors, e.g. at least the set of basis code vectors b_0 , is associated with at least two scale representatives, the same set of basis code vectors can be used to construct code vectors of the at least one set of code vectors associated with a first scale representative and to construct code vectors of the at least one set of code vectors associated with at least one further scale representative.
- 15

It is possible to determine, for each set of basis code vectors of a plurality of sets of basis code vectors, a potential basis code vector for encoding an input vector in other ways.

- 20 For example determining a code vector for encoding the input vector from a subset of code vectors is based on a determined distortion metric or distance, or error value.
- 25 In such examples a scale representation of the plurality of scale representations is selected.

Furthermore the determined potential basis code vector of a set of basis code vectors associated with the selected scale representation is selected.

- 30 A code vector may then be determined based on the selected potential basis code vector and on the selected scale representation, wherein this determining

of a code vector may be performed as described with respect to the method described herein.

In some examples based on the determined code vector and the input vector, a 5 distortion metric is determined. For instance, said distortion metric may be based on any kind of suitable distance between the determined code vector and the input vector. As an example, a Hamming distance or an Euclidian distance or any other distance may be used. As an example, determining the code vector may be omitted and the distortion metric may be calculated by inherently 10 considering the respective code vector associated with the selected scale representation and the set of basis code vectors associated with this selected scale representation.

For instance, if $c_{x,z,y} = [c_{x,z,y,0}, c_{x,z,y,1}, \dots, c_{x,z,y,n-1}]$ represents the code vector 15 determined in step 430 and $i = [i_0, i_1, \dots, i_{n-1}]$ represents the input vector, a distance d may be calculated based on

$$d = \sum_{k=0}^{n-1} (i_k - c_{x,z,y,k})^2.$$

20 This distance d according to the equation above may be replaced with distance d' calculated based on

$$d' = \sum_{k=0}^{n-1} c_{x,z,y,k}^2 - 2 \sum_{k=0}^{n-1} i_k \cdot c_{x,z,y,k}$$

25 Or, as another example, in case the distortion metric is determined based on a weighting function, distance d according to equation above may be amended as follows:

$$d_w = \sum_{k=0}^{n-1} w_k \cdot (i_k - c_{x,z,y,k})^2,$$

wherein w_k represent weighting factors of the weighting function.

5 Accordingly, distance d' according to the equation above may be weighted by means of the weighting function in the following way:

$$d'_w = \sum_{k=0}^{n-1} w_k \cdot c_{x,z,y,k}^2 - 2 \sum_{k=0}^{n-1} w_k \cdot i_k \cdot c_{x,z,y,k}$$

10 For instance, the distortion metric d , or d' , or d_w , or d'_w may be stored, if it is the first determined distortion metric, or it may be compared with a stored distortion metric, wherein the stored distortion metric is replaced if the newly determined distortion metric is better than the stored distortion metric. Furthermore, the code vector associated with the stored distortion metric may be stored or an
15 identifier of this code vector may be stored.

Then for example the operation can checked whether there are any further sets of basis code vectors associated with the selected scale representation. If yes, then the determined potential basis code vector of this further set of basis code
20 vectors associated with the selected scale representation is selected. If no there is a check made against further scale representation of the plurality of scale representations.

If there is a further scale representation of the plurality of scale representations, 25 then the further scale representation is selected, otherwise the code vector associated with the best distance metric may be selected for encoding the input vector.

For instance, where the sets of basis code vectors may represent leader classes, wherein each leader class comprises a different leader vector and permutations of said leader vector. Thus, the leader vector and the permutations of said leader vector may represent the basis code vectors of the 5 respective set of basis code vectors. As an example, a leader vector is an n -dimensional vector (with n denoting an integer number), whose (positive) components are ordered (e.g. decreasingly). The leader class corresponding to the leader vector then consists of the leader vector and all vectors obtained through all the signed permutations of the leader vector (with some possible 10 restrictions).

A union of leader classes may be defined by the sets of basis code vectors associated with the same scale representation of the plurality of scale representations and the respective scale representation. For instance, a union 15 of leader classes may be associated with a set of code vectors obtained by means of scaling the basis code vectors of the associated step of basis code vectors with the scale representative.

Such a union of leader classes may be considered as a truncation. Thus, if the 20 plurality of scale representations are n scale representations, n unions of leader classes may be defined, wherein each union of leader class is defined by means of the respective scale representation and the sets of basis code vectors associated with the respective scale representation.

25 Accordingly, the plurality of scale representations and the plurality of sets of basis code vectors may define a plurality of union of leader classes thereby defining a codebook, wherein, as an example, each union of leader classes may be considered as a union of scaled leader classes.

30 Codebooks used within these speech and audio codecs may for instance be based on lattice structures, as described in reference "Multiple-scale leader-lattice VQ with application to LSF quantization" by A. Vasilache, B. Dumitrescu

and I. Tabus, Signal Processing, 2002, vol. 82, pages 563-586, Elsevier, which is incorporated herein in its entirety by reference. For instance, a D10+ lattice may be considered for quantization, but any other well-suited lattice quantization may also be considered.

5

For instance the sets of basis code vectors are leader classes, wherein each leader class comprises a different leader vector and permutations of said leader vector, and wherein each leader vector represents an n-dimensional vector comprising n absolute values arranged in a descending or an ascending order.

10

The leader vector l of the respective set of basis code vectors b_x may be represented by $l=[l_0, l_1, \dots, l_{n-1}]$, wherein l_0, l_1, \dots, l_{n-1} are absolute values. In case of a descending order l_0 represents the 1-highest value, l_1 represents the 2-highest value and l_{n-1} represents the n-highest value. In case of an ascending order l_0 represents the 1-lowest value, l_1 represents the 2-lowest value and l_{n-1} represents the n-lowest value.

The value l_{k-1} of the respective leader vector, which represents the value at k th position in the respective leader vector, can be assigned to a position in the 20 potential basis code vector which corresponds to the position of the k -highest absolute value (in case of a descending ordered leader vector) or to the position of the k -lowest absolute value (in case of an ascending ordered leader vector) in the input vector. For instance, this position may be denoted as position m . As an example, the potential basis code vector may be represented by $p=[p_0, p_1, \dots, p_{n-1}]$.

25 For instance, as a non-limiting example, an exemplary input vector may be
30 $l=[-2.4, 5.0, -1.3, 0.2]$, wherein the corresponding absolute valued input vector
may be

$la=[2.4, 5.0, 1.3, 0.2]$.

In case of the descending order of the leader vector, the value in position k of the leader vector, i.e. value l_{k-1} , is assigned to a position in the potential basis code vector which corresponds to the position of the k -highest absolute value in the input vector. For instance, starting with the first position represented by counter $k=1$, the position of the 1-highest absolute value in the input vector is position $m=2$, since value 5.0 is the 1-highest value in the absolute valued input vector and is located in position $m=2$, i.e. i_1 . Accordingly, value l_0 is assigned to the position $m=2$ in the potential basis code vector, i.e. $p_1=l_0$ may hold.

10

Furthermore, the sign (+ or -) of the assigned value in the potential basis code vector p_{m-1} is set in accordance with the sign of the value of the input vector associated with the k -highest absolute value. Accordingly,

15 $p_{m-1} = l_{k-1} \cdot \text{sign}(i_{m-1})$

may hold.

Thus, in the non-limiting example of an exemplary input vector $i=[-2.4, 5.0, -1.3, 0.2]$, $p_1=l_0$ may hold since value $i_1=5.0$ has a positive sign.

The position counter k may be incremented, and it may be checked whether there is another value in the leader vector, i.e. whether $k \leq n$ holds.

25 If yes, the method proceeds and in the non-limiting example, with respect to position $k=2$, value 2.4 at position $m=1$ represents the 2-highest (k -highest) absolute value in the input vector. Thus,

$$P_0 = l_1 \cdot \text{sign}(i_0) = -l_1$$

30

may hold for assigning l_1 with the respective sign, since value $i_0=-2.4$ in the input vector has a negative sign.

In this way, for the non-limiting example, the loop may iterate through the positions of the leader vector in the following way:

5 $k=3 \rightarrow m=3 \rightarrow P_2 = l_2 \cdot \text{sign}(l_2) = -l_2$; and
k=4 $\rightarrow m=4 \rightarrow P_3 = l_3 \cdot \text{sign}(l_3) = +l_3$

Accordingly, the respective potential code vector obtained by the example method may result in $p=[-l_1, l_0, -l_2, l_3]$ in case of the descending ordered
10 respective leader vector l .

If the leader vector l is ordered in an ascending way, then the method described above may be performed with m representing the position of the k -lowest value in the absolute valued input vector, wherein $p_{m-1} = l_{m-1} \cdot \text{sign}(l_{m-1})$ may hold.

15 The obtained potential code vector p is associated with the respective set of basis code vectors b_x , wherein l represents the leader vector of this respective set of basis code vectors. For instance, with respect to the example process of determining a code vector based on a basis code vector $b_{x,y,l}$ and scale
20 representative s_z and described above, the potential code vector p represents the nearest basis code vector $b_{x,y}$ of the set of basis code vectors b_x with respect to the input vector, wherein the absolute valued input vector is used for determining the potential code vector of a respective set of basis code vectors and wherein the sign of each value $b_{x,y,k-1}$ at the k th position of the determined
25 nearest basis code vector $b_{x,y}$ is assigned with the sign of the respective value l_k at the k th position of the input vector l , wherein $0 < k \leq n$ holds.

Thus, this nearest basis code vector $b_{x,y}$ representing the potential code vector p can be used for determining a code vector $c_{x,z,y}$ based on the nearest basis
30 code vector $b_{x,y}$ and based on a respective scale representative s_z , as described above.

To each truncation a different scale representative is assigned (e.g. through training), e.g.:

float scale[] = {0.8, 1.2, 2.7};

5

Accordingly, for instance, a first set of code vectors of a plurality of code vectors of the codebook is defined by the first truncation scaled by the first scale representation 0.8, a second set of code vectors of the plurality of code vectors of the codebook is defined by the second truncation scaled by the second scale

10 representation 1.2, and a third set of code vectors of the plurality of code vectors of the codebook is defined by the third truncation scaled by the third scale representation 2.7, the codebook having a multiple scale lattice structure.

15 As an example, the search in the multiple scale lattice structure may be seen as having two phases: the first one may compute a potential code vector for each leader class, i.e. for each set of basis code vectors, and the second one may calculate the distortion only for the potential codevectors.

20 For instance, an absolute value function may be applied to the input vector i such that absolute input vector ia comprises the absolute values of the vector i , and then the absolute input vector may be sorted in an descending (or, alternatively, in an ascending) order.

25 As an example, an index representation may contain representatives indicating the indexes of each input vector i in the descendingly (or ascendingly) ordered absolute valued vector. For instance, said index representation may be an integer array 'indx'.

30 For example, if the input vector is [-2.4 5.0 -1.3 0.2], the absolute valued vector is [2.4 5.0 1.3 0.2] and the 'indx' array is [1 0 2 3]. Since the leader vectors may be descendingly ordered, during the nearest neighbour search algorithm, the

first value of the leader vector may be assigned on the position corresponding to the highest absolute value component of the input vector and so on.

In the following non-limiting example, 'idx_lead_max' is the maximum number of

5 leader classes out of all truncations, which may correspond to X, in this example may be is 9. Accordingly 9 sets of basis code vectors are defined by means of the 9 leader classer, wherein the nth leader class is defined by &pl[n-1].

10 For instance, the array 'sign' may store the signs of the input vector components.

/ First part of the search: compute all potential codevectors */*

```

pl_crt = &pl[0]; /* pl contains the leader vectors */
15   for (u=0;u<idx_lead_max;u++)
{
    for(j=0;j<LATTICE_DIM;j++, pl_crt++)
    {
        l_crt = indx[j];
        if ( (*pl_crt) > 0. )
        {
            cv_pot[u][l_crt] = (*pl_crt)*(float)sign[j][l_crt];
        }
        else
        {
            cv_pot[u][l_crt] = 0.0f;
        }
    }
}

```

30

The outer loop defined by counter u may be considered to associate each each u with a respective leader vector. Thus, in accordance with counter u, a corresponding set of basis code vectors is selected by means of the outer loop,

since each leader vector corresponds to a different set of basis code vectors of the plurality of basis code vectors.

5 The inner loop defined by integer value j may be considered to determining a potential basis code vector associated with the selected set of basis code vectors, j_crt indicating the position of the $(j+1)$ highest absolute value in the input vector.

10 Thus, the different potential basis code vectors cv_pot are determined by means of this exemplary first part of the search.

The second part of the search may be used for determining a code vector for encoding the input vector from a subset of code vectors.

```
15  /* Second part of the search */
    for(l=0;l<no_scales;l++)
    {
        s = scale[l];
        s2 = s*s;
20        for(k=0;k<LATTICE_DIM;k++)
        {
            ws1[k] = w[k]*s*2.0f*in[k];
            ws2[k] = w[k]*s2;
        }
25        for(j=0;j<no_leaders[l];j++)
        {
            tmp_dist = 0.0f;
            for(k=0;k<LATTICE_DIM;k++)
            {
                s = cv_pot[j][k];
                tmp_dist += (ws2[k]*s-ws1[k])*s;
            }
            if (tmp_dist < min_dist)
```

```
    {
        min_dist = tmp_dist;
        best_scale = i;
        best_idx = k;
    }
}
```

10 The outer loop may be defined by counter *i*, wherein *i* is issued to select one scale representation *scale[i]* of the plurality of scale representations.

LATTICE_DIM defines the length of the code vectors which may correspond to the length of the input vector to be encoded.

15 Afterwards, the values *ws1[k]* and *ws2[k]* for each *k* in $(0, \dots, \text{LATTICE_DIM})$ are calculated, which may be considered to be that part of the distortion metric (X3) which is independent of potential basis code vector. The value *w[k]* represents the value of the weighting function for each *k*.

20 The example code shown above further has an inner *j* loop “*for(j=0;j<no_leaders[i];j++)*”, wherein *no_leaders[i]* defines the set of leader vectors associated with the selected scale representative *scale[i]*, i.e. *no_leaders[i]* may correspond to *n_z* representing the number of sets of basis code vectors associated with the respective scale representative *scale[i]*, and

25 thus this loop iterates through each set of leader vectors associated with the selected scale representative *scale[i]*, wherein for leader vector of this set of leader vectors one potential basis code vector *cv_pot* has been determined. Thus, for instance, this loop iteratively selects each potential basis code vector *cv_pot* the set of basis code vectors associated with the selected scale

30 representation, wherein *cv_pot[j]* may represent the respective *j*th basis code vector of this set of basis code vectors.

For each of these basis code vectors and the selected scale representative, the respective distortion metric for the code vector being associated with the respective basis code vector and the selected scale representative may be determined, e.g. based on distortion metric in the following way:

5

$$d = \sum_{k=0}^{n-1} (ws2[k] \cdot cv_pot[j][k] - ws1[k]) \cdot cv_pot[j][k]$$

The distortion metric having the lowest value is determined to represent the best distortion metric, wherein the code vector associated with this distortion

10 metric code vector may be used for encoding the input vector. For instance, this code vector may be defined by the best scale representative and the best potential basis code vector of the set of potential basis code vectors.

15 The embodiments described herein reduce the complexity of the vector quantization by not computing the potential codevector array *cv_pot*, but employing the absolute value sorted version of the input vector and determining or generating the distortion calculation in a suitable transposed space.

20 In some embodiments the lattice vector quantizer comprises as input vector sorter 402. The input vector sorter 402 or suitable means for sorting the input vector can be configured to receive the input vector.

The operation of receiving the input vector is shown in Figure 6 by step 501.

25 The lattice vector quantizer and input vector sorter 402 is configured to sort the input vector into an absolute value descending order (it would be understood that in some embodiments the sorting can be performed in an absolute value ascending order with suitable changes to the following operations).

30 Thus for example if the input vector is

$$I = [-2.4 \ 5.0 \ -1.3 \ 0.2],$$

the absolute valued vector is

$$\text{absi} = [2.4 \ 5.0 \ 1.3 \ 0.2],$$

5 the sorted absolute valued vector which is defined here as

$$\text{cv_pot1} = [5.0 \ 2.4 \ 1.3 \ 0.2]$$

and the sorting permutation 'indx' = [1 0 2 3].

10 The sorting of the input vector is shown in Figure 6 by step 503.

The input vector sorter can then pass the sorted vector and sorting permutation to the code vector determiner 403.

15 In some embodiments the lattice vector quantizer 453 comprises a potential code vector determiner 401. The potential code vector determiner or suitable means for determining a potential code vector is configured to store or generate the leader classes used to generate the codevectors.

20 For instance the leader classes may be defined as (in Q1 value, in other words multiplied by 2)

```
const Word16 pl_fx[] = // Q1 vectors in first layers
25  {2, 2, 0, 0, 0, 0, 0, 0,
      1, 1, 1, 1, 1, 1, 1, 1,
      2, 2, 2, 2, 0, 0, 0, 0,
      4, 0, 0, 0, 0, 0, 0, 0,
      3, 1, 1, 1, 1, 1, 1, 1, // 5.
      2, 2, 2, 2, 2, 2, 0, 0,
      4, 2, 2, 0, 0, 0, 0, 0,
      3, 3, 1, 1, 1, 1, 1, 1,
      2, 2, 2, 2, 2, 2, 2, 2,
      4, 2, 2, 2, 2, 0, 0, 0, // 10.
      4, 4, 0, 0, 0, 0, 0, 0,
```

	3, 3, 3, 1, 1, 1, 1,	
	5, 1, 1, 1, 1, 1, 1,	
	4, 2, 2, 2, 2, 2, 0,	
	4, 4, 2, 2, 0, 0, 0,	11/11
5	6, 2, 0, 0, 0, 0, 0,	
	3, 3, 3, 3, 1, 1, 1,	
	5, 3, 1, 1, 1, 1, 1,	
	4, 4, 2, 2, 2, 2, 0, 0,	
	4, 4, 4, 0, 0, 0, 0,	11/230
10	6, 2, 2, 2, 0, 0, 0,	
	3, 3, 3, 3, 1, 1, 1,	
	5, 3, 3, 1, 1, 1, 1,	
	4, 4, 2, 2, 2, 2, 2, 2,	
	4, 4, 4, 2, 2, 0, 0, 0,	11/23
15	6, 2, 2, 2, 2, 2, 0, 0,	
	6, 4, 2, 0, 0, 0, 0, 0,	
	3, 3, 3, 3, 3, 1, 1,	
	5, 3, 3, 3, 1, 1, 1, 1,	
	5, 5, 1, 1, 1, 1, 1, 1,	11/23
20	7, 1, 1, 1, 1, 1, 1, 1,	
	4, 4, 4, 2, 2, 2, 2, 0,	
	4, 4, 4, 4, 0, 0, 0, 0,	
	6, 2, 2, 2, 2, 2, 2, 2,	
	6, 4, 2, 2, 2, 0, 0, 0, 0,	11/230

These leader classes can in some embodiments be passed to the code vector determiner 403.

30 In some embodiments the lattice vector quantizer 453 comprises a code vector determiner 403. The code vector determiner 403 or suitable means for determining a code vector can in some embodiments receive the leader classes and also the sorted input vector and permutation vector. The code vector determiner can then from these values determine the output code vector associated with the input vector.

Where the distance to be determined is a weighted Euclidean distance then in some embodiments the weights are transposed according to the permutation vector and an intermediary input vector produce is generated. It would be understood that in some embodiments the weights are uniform or the weighting operation is optional where the unweighted Euclidean distance is employed.

5 An example of this can be shown by the following code

```
/* calculate intermediary product between transposed weights and sorted input
10 vector */
for (j=0;j<LATTICE_DIM;j++)
{
    w_transp[j] = w[indx[j]];
    wx[j] = w_transp[j]*cv_pot[j];
15 }
```

The operation of transposing and applying weights to generate an intermediary product based on the sorted input vector and the transposed weights is shown in Figure 6 by step 505

20 In some embodiments the code vector determiner can determine distance components sum1 and sum2 for a first scale value scale[0].

This operation can be divided into the steps of:

25 Firstly, initialising the scale and square of the scale values for a first scale value scale[0].

The operation of initialising the scale and square of the scale values are shown

30 in Figure 6 by step 506.

Secondly, selecting a leader vector from the leader classes matrix. This is shown in the above matrix example as the matrix `pl_crt`.

The operation of selecting a leader vector is shown in Figure 6 by step 507.

5

Thirdly generating intermediary distance values `sum1` and `sum2` based on intermediary values and the selected leader vector.

The operation of generating intermediary distance values based on the selected

10 leader vector is shown in Figure 6 by step 509.

Fourthly, checking the parity conditions where the leader vector does not reach the 7th position and correcting the `sum1` value where the number of minus signs in the input vector differ from the constraint given in the leader class parity.

15

The operation of checking the parity conditions where the leader vector does not reach the 7th position and correcting the `sum1` value where the number of minus signs in the input vector differ from the constraint given in the leader class parity is shown in Figure 6 by step 511.

20

Fifthly, determining the distance or error value from the `sum1` and `sum2` values and then where the current leader vector distance is the smallest indicating the index of the smallest vector.

25 The operation of determining the distance for the leader vectors is shown in Figure 6 by step 513.

The operation can then loop round until all of the leader vectors have been selected.

30

The operation of checking whether all leader vectors have been selected and looping back where not all of the leader vectors have been selected is shown in Figure 6 step 514.

5 These steps can be shown in the following code

```

for(j=0;j<no_leaders[0];j++)
{
10    sum1[j] = 0;
    sum2[j] = 0;
    l = 0;
    while(l<LATTICE_DIM-1)
    {
15        p = *pl_crt;
        if (p)
        {
            sum1[j] = sum1[j] + wx[l] * p;
            sum2[j] = sum2[j] + w_transp[j]*p*p;
20        pl_crt++;
        l++;
        }
        else
        {
25            pl_crt += LATTICE_DIM-1;
            l = LATTICE_DIM;
        }
    }
    if (l - LATTICE_DIM+ 1 ==0)
30        /* if it went up to 7th position, some leaders
       have zeros at the end, so no need for them to check the
       parity, because they have null-parity */
        p = *pl_crt;
        if ( pl_par_fx[j] ) /* if non-zero parity */
35        {
            if ( sig -pl_par_fx[j] != 0 ) /* if number
               of minus signs in the input vector different from the
               constraint given by the leader class parity */
            {
40                sum1[j] = sum1[j] - wx[l]* p; /* here is
               subtraction */
                sum2[j] = sum2[j] + w_transp[l] *p*p;
                pl_crt++;
            }
        }
45    else
}

```

```

5
{
    sum1[j] = sum1[j] + wx[l] * p;
    sum2[j] = sum2[j] + w_transp[l]*p*p;
    pl_crt++;
}
10
else
{
    sum1[j] = sum1[j] + wx[l]* p;
    sum2[j] = sum2[j] + w_transp[l] *p*p;
    pl_crt++;
}
15
tmp_dist = sum2[j]*s2 -sum1[j]*s;
if (tmp_dist < min_dist )
{
    min_dist = tmp_dist;
    best_idx = j;
}
20
} /* end of j loop */

```

Then in some embodiments the code vector determiner can be configured to use the sum1 and sum2 values to determine distortion distances for other scales. A similar operation of checking for a 'best' scale value is further made.

The operation of determining distortion distances for other scales is shown in Figure 6 by step 515.

30 The operation of determining the distortion distance can for the other scales using the sum1 and sum2 values can be implemented using the following example code

```

35
for(k=1;k<no_scales;k++)
{
    s = scale[k];
    s2 = s*s;
    /* and now use the sum1, sum2 values calculated above
    to calculate distortion for the other scales */
40
    for(j=0;j<no_leaders[j];j++)
    {
        tmp_dist = sum2[j]*s2 -sum1[j]*s;
        if (tmp_dist < min_dist )

```

```

    {
        min_dist = tmp_dist;
        best_scale = k;
        best_idx = j;
5      }
    ...
}

```

Furthermore in some embodiments the code vector determiner can be
10 configured, once the best leader class and best scale are found, to calculate
the resulting codevector 'cv_out'.

The operation of performing a reverse transpose to calculate the codevector is
shown in Figure 6 by step 517.

15 In some embodiments the operation of calculating the codevector can be
implemented by the following example code.

```

/* inverse permutation */
20  for(j=0;j<LATTICE_DIM;j++)
{
    id[indx[j]] = j;
}
25  for(j=0;j<LATTICE_DIM;j++)
{
    cv_out[j] = sign[j]
*pl_fx[best_idx*LATTICE_DIM+id[j]]));
}
30  if(pl_par_fx[best_idx])
{
    if ( sig -pl_par_fx[best_idx] != 0 )
    {
        cv_out[smallest] = -cv_out[smallest];
    }
}
35

```

In some embodiments the calculation of the variables sum1 and sum2 is done
up to the number of leaders from the first truncation (no_leaders[0]), meaning

that the number of leaders should be decreasingly ordered and their corresponding scales ordered accordingly.

In such embodiments an additional complexity reduction is produced because

5 the maximum number of leaders for one structure need not be computed, but it is known to be on the first position.

It would be understood that most of the complexity reduction comes from the fact that only the winning leader vector has to be transposed, not all of them.

10 The calculation is done on positive values (both leader vector and input vector are in absolute values) which is ok as long as the input vector component and the quantized one have the same sign.

A difference in sign intervenes when there is a parity constraint (odd or even 15 number of negative components) in the considered leader vector and this constraint is not respected by the input vector. In this case the sign of quantized value of the smallest input vector components has its sign flipped. The smallest input vector component corresponds to the last component in the transposed space. This is why the first loop for calculating sum1 and sum2 is 20 "while(l<LATTICE_DIM-1)". In the real, non-transposed space this corresponds to smallest = indx[LATTICE_DIM-1]. LATTICE_DIM is the dimension of the considered lattice.

25 Although the above examples describe embodiments of the application operating within a codec within an apparatus 10, it would be appreciated that the invention as described below may be implemented as part of any audio (or speech) codec, including any variable rate/adaptive rate audio (or speech) codec. Thus, for example, embodiments of the application may be implemented in an audio codec which may implement audio coding over fixed or wired 30 communication paths.

Thus user equipment may comprise an audio codec such as those described in embodiments of the application above.

It shall be appreciated that the term user equipment is intended to cover any

5 suitable type of wireless user equipment, such as mobile telephones, portable data processing devices or portable web browsers.

Furthermore elements of a public land mobile network (PLMN) may also comprise audio codecs as described above.

10

In general, the various embodiments of the application may be implemented in hardware or special purpose circuits, software, logic or any combination thereof.

For example, some aspects may be implemented in hardware, while other aspects may be implemented in firmware or software which may be executed

15 by a controller, microprocessor or other computing device, although the invention is not limited thereto. While various aspects of the application may be illustrated and described as block diagrams, flow charts, or using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as

20 non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

The embodiments of this application may be implemented by computer software

25 executable by a data processor of the mobile device, such as in the processor entity, or by hardware, or by a combination of software and hardware. Further in this regard it should be noted that any blocks of the logic flow as in the Figures may represent program steps, or interconnected logic circuits, blocks and functions, or a combination of program steps and logic circuits, blocks and

30 functions.

The memory may be of any type suitable to the local technical environment and may be implemented using any suitable data storage technology, such as semiconductor-based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory. The data processors may be of any type suitable to the local technical environment, and may include one or more of general purpose computers, special purpose computers, microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASIC), gate level circuits and processors based on multi-core processor architecture, as non-limiting examples.

10

Embodiments of the application may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

Programs, such as those provided by Synopsys, Inc. of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre-stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

25 As used in this application, the term 'circuitry' refers to all of the following:

(a) hardware-only circuit implementations (such as implementations in only analogue and/or digital circuitry) and

(b) to combinations of circuits and software (and/or firmware), such as: (i) to a combination of processor(s) or (ii) to portions of processor(s)/software (including digital signal processor(s)), software, and memory(ies) that work together to cause an apparatus, such as a mobile phone or server, to perform various functions and

(c) to circuits, such as a microprocessor(s) or a portion of a microprocessor(s), that require software or firmware for operation, even if the software or firmware is not physically present.

5 This definition of 'circuitry' applies to all uses of this term in this application, including any claims. As a further example, as used in this application, the term 'circuitry' would also cover an implementation of merely a processor (or multiple processors) or portion of a processor and its (or their) accompanying software and/or firmware. The term 'circuitry' would also cover, for example and if 10 applicable to the particular claim element, a baseband integrated circuit or applications processor integrated circuit for a mobile phone or similar integrated circuit in server, a cellular network device, or other network device.

15 The foregoing description has provided by way of exemplary and non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this 20 invention will still fall within the scope of this invention as defined in the appended claims.

Claims:

1. A method comprising:
 - generating at least one vector of parameters defining at least one audio signal;
 - 5 sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters;
 - selecting from a list of leader classes at least one potential code vector;
 - 10 determining a distance between the at least one potential code vector and the at least one ordered vector of parameters;
 - determining at least one leader class associated with a potential code vector which generates the smallest associated distance;
 - transposing the at least one leader class to generate an output lattice quantized codevector.
2. The method as claimed in claim 1, further comprising:
 - selecting at least one scale factor;
 - wherein determining a distance between the at least one potential code vector and the at least one ordered vector of parameters further comprises determining a distance based on the at least one scale factor;
 - 20 wherein determining at least one leader class associated with a potential code vector which generates the smallest associated distance further comprises determining an output scale factor associated with a potential code vector and scale factor which generates the smallest associated distance;
 - 25 wherein transposing the at least one leader class to generate an output lattice quantized codevector comprises applying the output scale factor to the output lattice quantized codevector.
- 30 3. The method as claimed in any of claims 1 and 2, wherein generating a first vector of parameters defining at least one audio signal comprises:
 - dividing the at least one audio signal into time frames;

determining a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

4. The method as claimed in any of claims 1 to 3, wherein sorting the at
5 least one vector of parameters according to an ordering of at least one vector
absolute tuples to generate an associated at least one ordered vector of
parameters comprises:

determining weights for a weighted distance determination;

10 sorting the weights based on the ordering of the at least one vector
absolute tuples to generate a sorted weight vector;

applying the sorted weight vector to the at least one ordered vector of
parameters.

5. The method as claimed in any of claims 1 to 4, wherein determining a
15 distance between the at least one potential code vector and the at least one
ordered vector of parameters comprises:

generating a first and second distance components based on a first scale
value, the first and second distance components associated with the distance
between the at least one potential code vector and the at least one ordered
20 vector of parameters;

correcting the sign of the first distance component where the potential
code vector fails a parity constraint test;

25 determining a distance between the at least one potential code vector
and the at least one ordered vector of parameters based on the first and second
distance components.

6. The method as claimed in claim 5, wherein determining a distance
between the at least one potential code vector and the at least one ordered
vector of parameters further comprises determining a distance between the at
30 least one potential code vector and the at least one ordered vector of
parameters based on the first and second distance components and further
scale values.

7. The method as claimed in any of claims 1 to 6, wherein determining at least one leader class associated with a potential code vector which generates the smallest associated distance comprises:

5 determining the index of the potential code vector which generates the smallest associated distance;

 determining the scale value of the potential code vector which generates the smallest associated distance by applying scale values to only the potential code vector which generates the smallest associated distance.

10

8. The method as claimed in claims 1 to 7, wherein transposing the at least one leader class to generate an output lattice quantized codevector comprises sorting the at least one leader class by the reverse ordering of at least one vector absolute tuples.

15

9. An apparatus comprising:

 means for generating at least one vector of parameters defining at least one audio signal;

20 means for sorting the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters;

 means for selecting from a list of leader classes at least one potential code vector;

25 means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters;

 means for determining at least one leader class associated with a potential code vector which generates the smallest associated distance;

 means for transposing the at least one leader class to generate an output lattice quantized codevector.

30

10. The apparatus as claimed in claim 9, further comprising:

 means for selecting at least one scale factor;

wherein the means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters further comprises means for determining a distance based on the at least one scale factor;

5 the means for determining at least one leader class associated with a potential code vector which generates the smallest associated distance further comprises means for determining an output scale factor associated with a potential code vector and scale factor which generates the smallest associated distance;

10 the means for transposing the at least one leader class to generate an output lattice quantized codevector comprises means for applying the output scale factor to the output lattice quantized codevector.

11. The apparatus as claimed in any of claims 9 and 10, wherein the means
15 for generating a first vector of parameters defining at least one audio signal
comprises:

means for dividing the at least one audio signal into time frames;
means for determining a vector of line spectral frequency parameters
associated with the at least one audio signal time frames.

20 12. The apparatus as claimed in any of claims 9 to 11, wherein the means
for sorting the at least one vector of parameters according to an ordering of at
least one vector absolute tuples to generate an associated at least one ordered
vector of parameters comprises:

25 means for determining weights for a weighted distance determination;
means for sorting the weights based on the ordering of the at least one
vector absolute tuples to generate a sorted weight vector;
means for applying the sorted weight vector to the at least one ordered
vector of parameters.

13. The apparatus as claimed in any of claims 9 to 12, wherein the means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters comprises:

means for generating a first and second distance components based on
5 a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered vector of parameters;

means for correcting the sign of the first distance component where the potential code vector fails a parity constraint test;

10 means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components.

14. The apparatus as claimed in claim 13, wherein the means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters further comprises means for determining a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components and further scale values.

20

15. The apparatus as claimed in any of claims 9 to 14, wherein the means for determining at least one leader class associated with a potential code vector which generates the smallest associated distance comprises:

means for determining the index of the potential code vector which generates the smallest associated distance;

means for determining the scale value of the potential code vector which generates the smallest associated distance by applying scale values to only the potential code vector which generates the smallest associated distance.

30 16. The apparatus as claimed in claims 9 to 15, wherein the means for transposing the at least one leader class to generate an output lattice quantized

codevector comprises means for sorting the at least one leader class by the reverse ordering of at least one vector absolute tuples.

17. An apparatus comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus at least to:

5 generate at least one vector of parameters defining at least one audio signal;

10 sort the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters;

15 select from a list of leader classes at least one potential code vector;

determine a distance between the at least one potential code vector and

15 the at least one ordered vector of parameters;

determine at least one leader class associated with a potential code vector which generates the smallest associated distance;

20 transpose the at least one leader class to generate an output lattice quantized codevector.

18. The apparatus as claimed in claim 17, further caused to:

select at least one scale factor;

25 wherein determining a distance between the at least one potential code vector and the at least one ordered vector of parameters further causes the apparatus to determine a distance based on the at least one scale factor;

wherein determining at least one leader class associated with a potential code vector which generates the smallest associated distance further causes the apparatus to determine an output scale factor associated with a potential code vector and scale factor which generates the smallest associated distance;

30 wherein transposing the at least one leader class to generate an output lattice quantized codevector causes the apparatus to apply the output scale factor to the output lattice quantized codevector.

19. The apparatus as claimed in any of claims 17 and 18, wherein generating a first vector of parameters defining at least one audio signal causes the apparatus to:

5 divide the at least one audio signal into time frames;
determine a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

20. The apparatus as claimed in any of claims 17 to 19, wherein sorting the 10 at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters causes the apparatus to:

determine weights for a weighted distance determination;
sort the weights based on the ordering of the at least one vector absolute 15 tuples to generate a sorted weight vector;
apply the sorted weight vector to the at least one ordered vector of parameters.

21. The apparatus as claimed in any of claims 17 to 20, wherein determining 20 a distance between the at least one potential code vector and the at least one ordered vector of parameters causes the apparatus to:

generate a first and second distance components based on a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered 25 vector of parameters;
correct the sign of the first distance component where the potential code vector fails a parity constraint test;
determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second 30 distance components.

22. The apparatus as claimed in claim 21, wherein determining a distance between the at least one potential code vector and the at least one ordered vector of parameters further causes the apparatus to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components and further scale values.

5 23. The apparatus as claimed in any of claims 17 to 22, wherein determining at least one leader class associated with a potential code vector which generates the smallest associated distance causes the apparatus to:

10 determine the index of the potential code vector which generates the smallest associated distance;

15 determine the scale value of the potential code vector which generates the smallest associated distance by applying scale values to only the potential code vector which generates the smallest associated distance.

24. The apparatus as claimed in claims 17 to 23, wherein transposing the at least one leader class to generate an output lattice quantized codevector causes the apparatus to sort the at least one leader class by the reverse ordering of at least one vector absolute tuples.

20 25. An apparatus comprising:

25 a vector generator configured to generate at least one vector of parameters defining at least one audio signal;

30 a lattice vector quantizer configured to sort the at least one vector of parameters according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters;

the lattice vector quantizer configured to select from a list of leader classes at least one potential code vector;

35 the lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters;

the lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance;

5 the lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector.

26. The apparatus as claimed in claim 25, further comprising:

the lattice vector quantizer configured to select at least one scale factor;

10 wherein the lattice vector quantizer configured to determining a distance between the at least one potential code vector and the at least one ordered vector of parameters further is configured to determine a distance based on the at least one scale factor;

the lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance further configured to determine an output scale factor 15 associated with a potential code vector and scale factor which generates the smallest associated distance;

the lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector further 20 configured to apply the output scale factor to the output lattice quantized codevector.

27. The apparatus as claimed in any of claims 25 and 26, wherein the apparatus further comprises a parameter determiner configured to divide the at

25 least one audio signal into time frames; and determine a vector of line spectral frequency parameters associated with the at least one audio signal time frames.

28. The apparatus as claimed in any of claims 25 to 27, wherein the lattice vector quantizer configured to sort the at least one vector of parameters 30 according to an ordering of at least one vector absolute tuples to generate an associated at least one ordered vector of parameters is further configured to:

determine weights for a weighted distance determination;

sort the weights based on the ordering of the at least one vector absolute tuples to generate a sorted weight vector;

apply the sorted weight vector to the at least one ordered vector of parameters.

5

29. The apparatus as claimed in any of claims 25 to 28, wherein the lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters is further configured to:

10 generate a first and second distance components based on a first scale value, the first and second distance components associated with the distance between the at least one potential code vector and the at least one ordered vector of parameters;

15 correct the sign of the first distance component where the potential code vector fails a parity constraint test;

determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and second distance components.

20 30. The apparatus as claimed in claim 29, wherein the lattice vector quantizer configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters further is configured to determine a distance between the at least one potential code vector and the at least one ordered vector of parameters based on the first and 25 second distance components and further scale values.

31. The apparatus as claimed in any of claims 25 to 30, wherein the lattice vector quantizer configured to determine at least one leader class associated with a potential code vector which generates the smallest associated distance is 30 further configured to:

determine the index of the potential code vector which generates the smallest associated distance;

determine the scale value of the potential code vector which generates the smallest associated distance by applying scale values to only the potential code vector which generates the smallest associated distance.

- 5 32. The apparatus as claimed in claims 25 to 31, wherein the lattice vector quantizer configured to transpose the at least one leader class to generate an output lattice quantized codevector is configured to sort the at least one leader class by the reverse ordering of at least one vector absolute tuples.
- 10 33. A computer program product for causing an apparatus to perform the method of any of claims 1 to 8.
34. An electronic device comprising apparatus as claimed in claims 9 to 32.
- 15 35. A chipset comprising apparatus as claimed in claims 9 to 32.

Figure 1

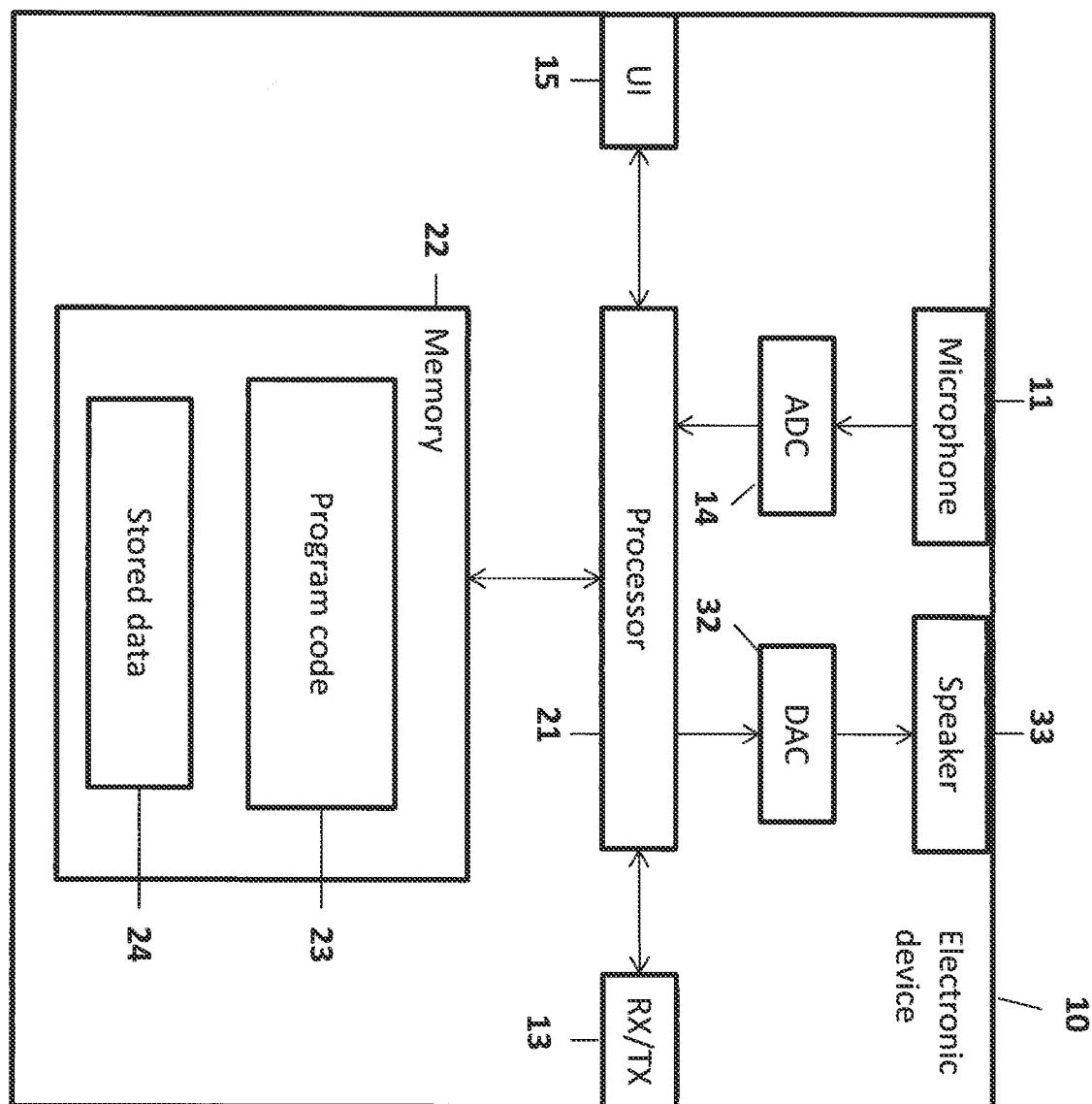


Figure 2

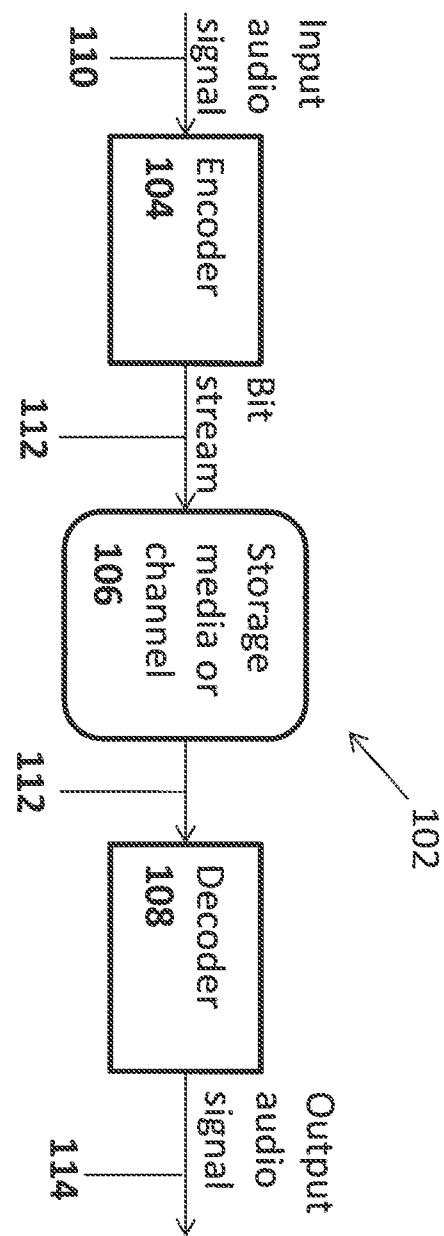


Figure 3

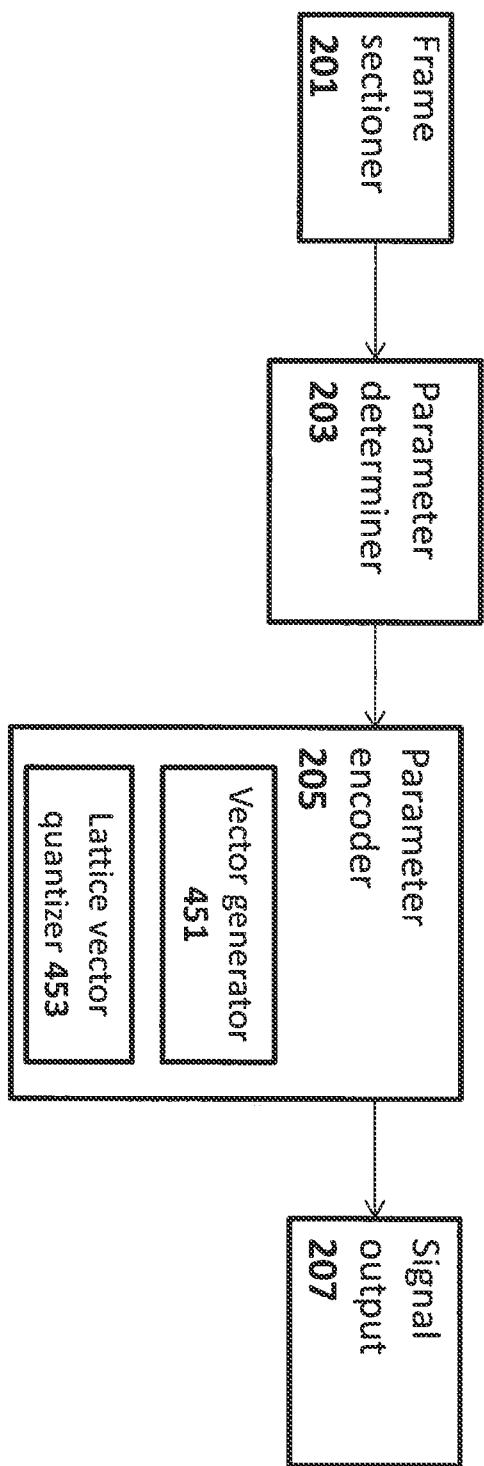


Figure 4

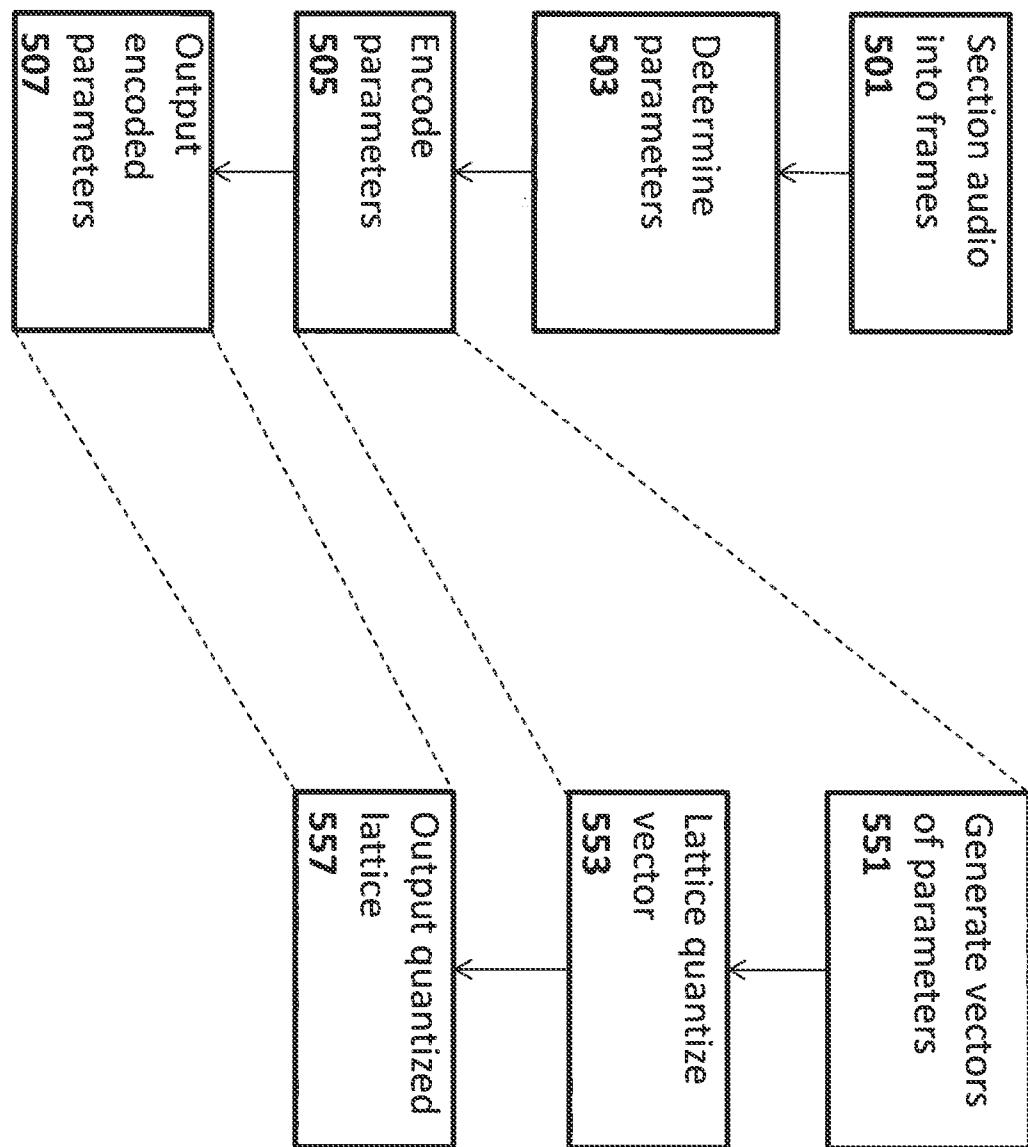


Figure 5

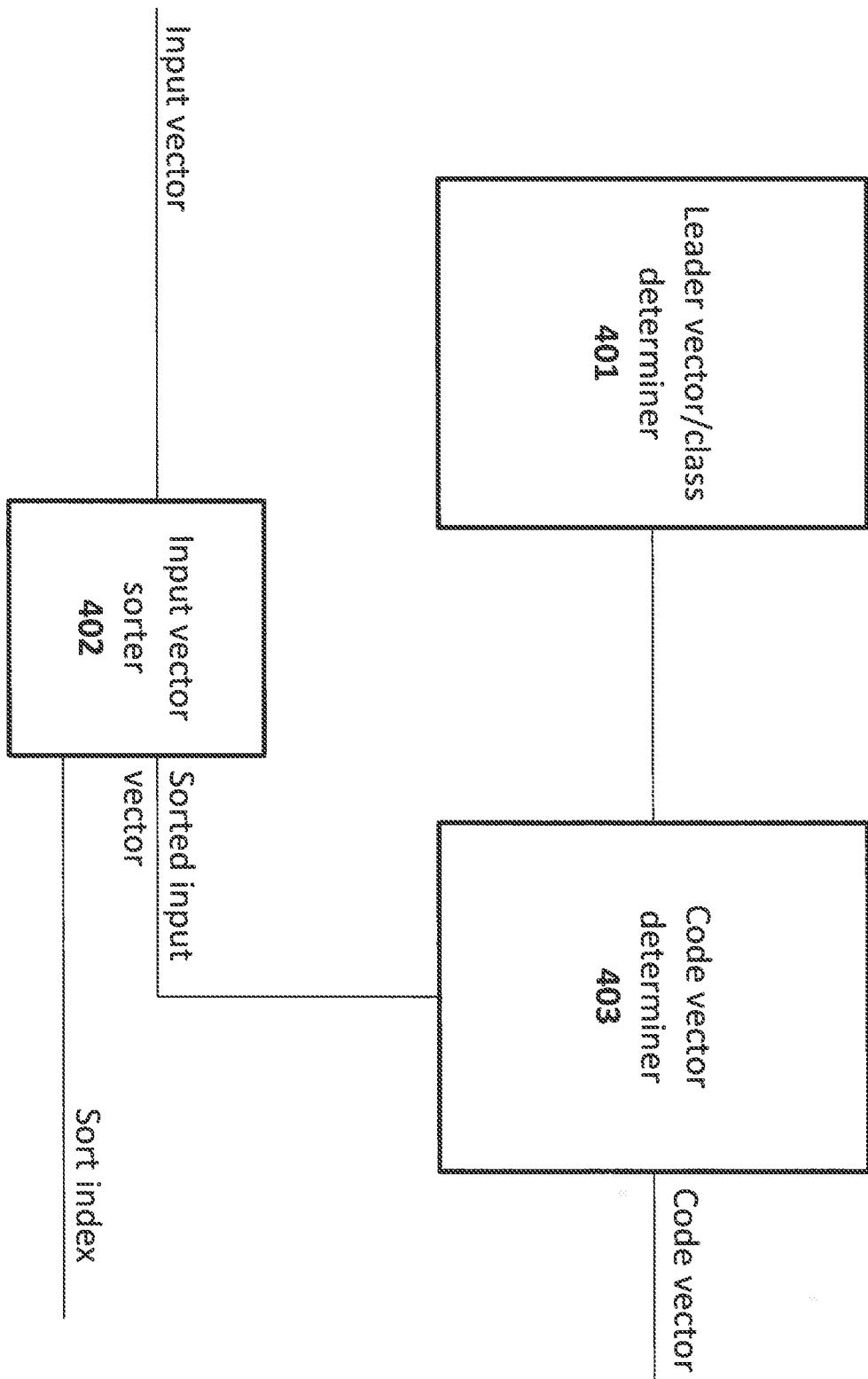
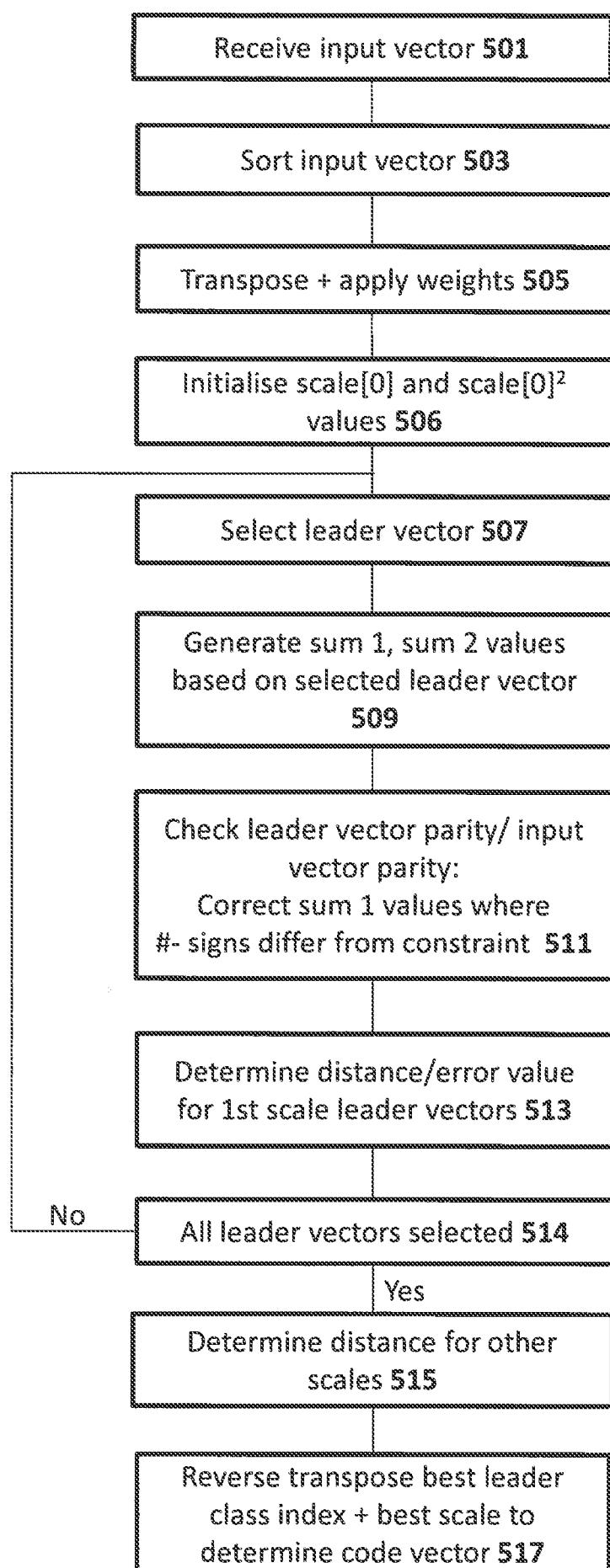


Figure 6



INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2013/061034

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G10L

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

SE, DK, FI, NO classes as above

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

EPO-Internal, PAJ, WPI data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2013005065 A1 (NOKIA CORP ET AL), 10 January 2013 (2013-01-10); whole document --	1-35
A	WO 2012069886 A1 (NOKIA CORP ET AL), 31 May 2012 (2012-05-31); whole document --	1-35
A	WO 2009127097 A1 (HUAWEI TECH CO LTD ET AL), 22 October 2009 (2009-10-22); whole document --	1-35
A	WO 2005083889 A1 (FRANCE TELECOM ET AL), 9 September 2005 (2005-09-09); whole document --	1-35



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

“E” earlier application or patent but published on or after the international filing date

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

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

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

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

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

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

“&” document member of the same patent family

Date of the actual completion of the international search

10-09-2014

Date of mailing of the international search report

11-09-2014

Name and mailing address of the ISA/SE

Patent- och registreringsverket

Box 5055

S-102 42 STOCKHOLM

Facsimile No. + 46 8 666 02 86

Authorized officer

Henrik Andersson

Telephone No. + 46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2013/061034

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 20050285764 A1 (BESSETTE BRUNO ET AL), 29 December 2005 (2005-12-29); whole document --	1-35
A	US 20090175550 A1 (TALEB ANISSE), 9 July 2009 (2009-07-09); whole document --	1-35
A	US 20100274556 A1 (SATO KAORU ET AL), 28 October 2010 (2010-10-28); whole document --	1-35
A	Vasilache A; Tabus I, "Indexing and entropy coding of lattice codevectors", 2001 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING. PROCEEDINGS. (ICASSP). SALT LAKE CITY, UT, MAY 7 - 11, 2001; [IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (ICASSP)], 20010507, 20010507 - 20010511, NEW YORK, NY : IEEE, US, ISBN 978-0-7803-7041-8 ; ISBN 0-7803-7041-4; whole document --	1-35
A	Nikneshan S; Khndani A K, "Efficient methods for the addressing/decoding of a lattice-based fixed-rate, entropy-coded vector quantizer", Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on Montreal, Que., Canada 8-12 June 1997, 19970608, 19970608, 19970612, New York, NY, USA, IEEE, US, ISBN 978-0-7803-3925-5 ; ISBN 0-7803-3925-8; whole document -- -----	1-35

Continuation of: second sheet

International Patent Classification (IPC)

G10L 19/038 (2013.01)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/IB2013/061034

WO	2013005065	A1	10/01/2013	CN	103636129	A	12/03/2014
				EP	2727106	A1	07/05/2014
				US	20140129216	A1	08/05/2014
WO	2012069886	A1	31/05/2012	US	20130246076	A1	19/09/2013
WO	2009127097	A1	22/10/2009	CN	102132494	A	20/07/2011
				EP	2274833	A1	19/01/2011
				US	8364476	B2	29/01/2013
				US	20110137645	A1	09/06/2011
WO	2005083889	A1	09/09/2005	CN	1906855	B	02/04/2014
				EP	1709743	A1	11/10/2006
				JP	2007523530	A	16/08/2007
				JP	4579930	B2	10/11/2010
				KR	20060129417	A	15/12/2006
				KR	101190875	B1	15/10/2012
				US	20070162236	A1	12/07/2007
				US	7680670	B2	16/03/2010
US	20050285764	A1	29/12/2005	AT	440409	T	15/09/2009
				AU	2003233723	A1	19/12/2003
				BR	0311336	A	15/03/2005
				CA	2388358	A1	30/11/2003
				CN	1659785	B	20/04/2011
				DE	60328892	D1	01/10/2009
				DK	1514355	T3	14/12/2009
				ES	2331876	T3	19/01/2010
				JP	4224021	B2	12/02/2009
				JP	2005528839	A	22/09/2005
				KR	20050008761	A	21/01/2005
				MX	PA04011841	A	26/07/2005
				NO	20045718	A	22/02/2005
				RU	2004138289	A	10/06/2005
				US	7106228	B2	12/09/2006
				WO	03103151	A1	11/12/2003
				ZA	200408780	A	30/08/2005
US	20090175550	A1	09/07/2009	CN	101292427	B	23/05/2012
				EP	1946447	A2	23/07/2008
				JP	2009509456	A	05/03/2009
				JP	4801160	B2	26/10/2011
				US	8340450	B2	25/12/2012
				WO	2007035148	A3	10/05/2007
US	20100274556	A1	28/10/2010	WO	2009090875	A1	23/07/2009