(19) DANMARK

(10) DK/EP 2684190 T3

(12) Oversættelse af europæisk patentskrift

Patent- og Varemærkestyrelsen

(51) Int.Cl.: ***G 10 L 19/028 (2013.01)*** *G 10 L 19/02 (2013.01)*

(45) Oversættelsen bekendtgjort den: **2016-02-22**

(80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2015-11-18**

(86) Europæisk ansøgning nr.: **11860593.0**

(86) Europæisk indleveringsdag: **2011-09-14**

(87) Den europæiske ansøgnings publiceringsdag: **2014-01-15**

(86) International ansøgning nr.: **SE2011051110**

(87) Internationalt publikationsnr.: **WO2012121638**

(30) Prioritet: **2011-03-10 US 201161451363 P**

(84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**

(73) Patenthaver: **Telefonaktiebolaget L M Ericsson (publ), , 164 83 Stockholm, Sverige**

(72) Opfinder: **GRANCHAROV, Volodya, Ankdammsgatan 29, SE-171 67 Solna, Sverige**
SVERRISSON, Sigurdur, Hallonvägen 54 2tr, SE-19635 Kungsängen, Sverige
NÄSLUND, Sebastian, Bokvägen 13, 16933 Solna, Sverige

(74) Fuldmægtig i Danmark: **Marks & Clerk (Luxembourg) LLP, 44 rue de la Vallée, B.P. 1775, L-1017 Luxembourg, Luxembourg**

(54) Benævnelse: **FYLDNING AF IKKE-KODEDE UNDERVEKTORER I TRANSFORMATIONSKODEDE LYDSIGNALER**

(56) Fremdragne publikationer:
EP-A1- 2 048 787
EP-A1- 2 234 104
WO-A1-00/11657
US-A1- 2003 233 234
US-A1- 2009 198 491
US-A1- 2009 299 738
US-A1- 2010 241 437
US-B1- 6 952 671
SANJEEV MEHROTRA ET AL: "Hybrid low bitrate audio coding using adaptive gain shape vector quantization", MULTIMEDIA SIGNAL PROCESSING, 2008 IEEE 10TH WORKSHOP ON, IEEE, PISCATAWAY, NJ, USA, 8 October 2008 (2008-10-08), pages 927-932, XP031356759, ISBN: 978-1-4244-2294-4

DESCRIPTION

TECHNICAL FIELD

[0001] The present technology relates to coding of audio signals, and especially to filling of non-coded sub-vectors in transform coded audio signals.

BACKGROUND

[0002] A typical encoder/decoder system based on transform coding is illustrated in Fig. 1.

[0003] Major steps in transform coding are:

A. Transform a short audio frame (20-40 ms) to a frequency domain, e.g., through the Modified Discrete Cosine Transform (MDCT).

B. Split the MDCT vector $X(k)$ into multiple bands (sub-vectors SV1, SV2, ...), as illustrated in Fig. 2. Typically the width of the bands increases towards higher frequencies [1].

C. Calculate the energy in each band. This gives an approximation of the spectrum envelope, as illustrated in Fig. 3.

D. The spectrum envelope is quantized, and the quantization indices are transmitted to the decoder.

E. A residual vector is obtained by scaling the MDCT vector with the envelope gains, e.g., the residual vector is formed by the MDCT sub-vectors (SV1,SV2, ...) scaled to unit Root-Mean-Square (RMS) energy.

F. Bits for quantization of different residual sub-vectors are assigned based on envelope energies. Due to a limited bit-budget, some of the sub-vectors are not assigned any bits. This is illustrated in Fig. 4, where sub-vectors corresponding to envelope gains below a threshold TH are not assigned any bits.

G. Residual sub-vectors are quantized according to the assigned bits, and quantization indices are transmitted to the decoder. Residual quantization can, for example, be performed with the Factorial Pulse Coding (FPC) scheme [2].

H. Residual sub-vectors with zero bits assigned are not coded, but instead noise-filled at the decoder. This is achieved by creating a Virtual Codebook (VC) from coded sub-vectors by concatenating the perceptually relevant coefficients of the decoded spectrum. The VC creates content in the non-coded residual sub-vectors.

I. At the decoder, the MDCT vector is reconstructed by up-scaling residual sub-vectors with corresponding envelope gains, and the inverse MDCT is used to reconstruct the time-domain audio frame.

[0004] A drawback of the conventional noise-fill scheme, e.g. as in [1], is that it in step H creates audible distortion in the reconstructed audio signal, when used with the FPC scheme.

[0005] US 2010/0241437 A1 discloses a method for perceptual spectral decoding, where an initial set of spectral coefficients is spectrum filled. The spectrum filling comprises noise filling of spectral holes by setting spectral coefficients in the initial set of spectral coefficients not being decoded from a binary flux equal to elements derived from decoded spectral coefficients. The set of reconstructed spectral coefficients of a frequency domain formed by the spectrum filling is converted into an audio signal of a time domain.

SUMMARY

[0006] A general object is an improved filling of non-coded residual sub-vectors of a transform coded audio signal.

[0007] Another object is generation of virtual codebooks used to fill the non-coded residual sub-vectors.

[0008] These objects are achieved in accordance with the attached claims.

[0009] A first aspect of the present technology involves a method of filling non-coded residual sub-vectors of a transform coded audio signal. The method includes the steps:

- Compressing actually coded residual sub-vectors.
- Rejecting compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion.
- Concatenating the remaining compressed residual sub-vectors to form a first virtual codebook.
- Combining pairs of coefficients of the first virtual codebook to form a second virtual codebook.
- Filling non-coded residual sub-vectors below a predetermined frequency with coefficients from the first virtual codebook.
- Filling non-coded residual sub-vectors above the predetermined frequency with coefficients from the second virtual codebook.

[0010] A second aspect of the present technology involves a method of generating a virtual codebook for filling non-coded residual sub-vectors of a transform coded audio signal below a predetermined frequency. The method includes the steps:

- Compressing actually coded residual sub-vectors.
- Rejecting compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion.
- Concatenating the remaining compressed residual sub-vectors to form the virtual codebook.

[0011] A third aspect of the present technology involves a method of generating a virtual codebook for filling non-coded residual sub-vectors of a transform coded audio signal above a predetermined frequency. The method includes the steps:

- Generating a first virtual codebook in accordance with the second aspect.
- Combining pairs of coefficients of the first virtual codebook.

[0012] A fourth aspect of the present technology involves a spectrum filler for filling non-coded residual sub-vectors of a transform coded audio signal. The spectrum filler includes:

- A sub-vector compressor configured to compress actually coded residual sub-vectors.
- A sub-vector rejecter configured to reject compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion.
- A sub-vector collector configured to concatenate the remaining compressed residual sub-vectors to form a first virtual codebook.
- A coefficient combiner configured to combine pairs of coefficients of the first virtual codebook to form a second virtual codebook.
- A sub-vector filler configured to fill non-coded residual sub-vectors below a predetermined frequency with coefficients from the first virtual codebook, and to fill non-coded residual sub-vectors above the predetermined frequency with coefficients from the second virtual codebook.

[0013] A fifth aspect of the present technology involves a decoder including a spectrum filler in accordance with the fourth aspect.

[0014] A sixth aspect of the present technology involves a user equipment including a decoder in accordance with the fifth aspect.

[0015] A seventh aspect of the present technology involves a low frequency virtual codebook generator for generating a low frequency virtual codebook for filling non-coded residual sub-vectors of a transform coded audio signal below a predetermined frequency. The low frequency virtual codebook generator includes:

- A sub-vector compressor configured to compress actually coded residual sub-vectors.
- A sub-vector rejecter configured to reject compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion.

- A sub-vector collector configured to concatenate the remaining compressed residual sub-vectors to form the low frequency virtual codebook.

[0016] An eight aspect of the present technology involves a high frequency virtual codebook generator for generating a high frequency virtual codebook for filling non-coded residual sub-vectors of a transform coded audio signal above a predetermined frequency. The low frequency virtual codebook generator includes:

- A low frequency virtual codebook generator in accordance with the seventh aspect configured to generate a low frequency virtual codebook.
- A coefficient combiner configured to combine pairs of coefficients of the low frequency virtual codebook to form the high frequency virtual codebook.

[0017] An advantage of the present spectrum filling technology is a perceptual improvement of decoded audio signals compared to conventional noise filling.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The present technology, together with further objects and advantages thereof, may best be understood by making reference to the following description taken together with the accompanying drawings, in which:

Fig. 1 is a block diagram illustrating a typical transform based audio coding/decoding system;

Fig. 2 is a diagram illustrating the structure of an MDCT vector;

Fig. 3 is a diagram illustrating the energy distribution in the sub-vectors of an MDCT vector;

Fig. 4 is a diagram illustrating the use of the spectrum envelope for bit allocation;

Fig. 5 is a diagram illustrating a coded residual;

Fig. 6 is a diagram illustrating compression of a coded residual;

Fig. 7 is a diagram illustrating rejection of coded residual sub-vectors;

Fig. 8 is a diagram illustrating concatenation of surviving residual sub-vectors to form a first virtual codebook;

Fig. 9A-B are diagrams illustrating combining of coefficients from the first virtual codebook to form a second virtual codebook;

Fig. 10 is a block diagram illustrating an example embodiment of a low frequency virtual codebook generator;

Fig. 11 is a block diagram illustrating an example embodiment of a high frequency virtual codebook generator;

Fig. 12 is a block diagram illustrating an example embodiment of a spectrum filler;

Fig. 13 is a block diagram illustrating an example embodiment of a decoder including a spectrum filler;

Fig. 14 is a flow chart illustrating low frequency virtual codebook generation;

Fig. 15 is a flow chart illustrating high frequency virtual codebook generation;

Fig. 16 is a flow chart illustrating spectrum filling;

Fig. 17 is a block diagram illustrating an example embodiment of a low frequency virtual codebook generator;

Fig. 18 is a block diagram illustrating an example embodiment of a high frequency virtual codebook generator;

Fig. 19 is a block diagram illustrating an example embodiment of a spectrum filler; and

Fig. 20 is a block diagram illustrating an example embodiment of a user equipment.

DETAILED DESCRIPTION

[0019] Before the present technology is described in more detail, transform based coding/decoding will be briefly described with reference to Fig. 1-7.

[0020] Fig. 1 is a block diagram illustrating a typical transform based audio coding/decoding system. An input signal $x(n)$ is forwarded to a frequency transformer, for example an MDCT transformer 10, where short audio frames (20-40 ms) are transformed into a frequency domain. The resulting frequency domain signal $X(k)$ is divided into multiple bands (sub-vectors SV1, SV2, ...), as illustrated in Fig. 2. Typically the width of the bands increases towards higher frequencies [1]. The energy of each band is determined in an envelope calculator and quantizer 12. This gives an approximation of the spectrum envelope, as illustrated in Fig. 3. Each sub-vector is normalized into a residual sub-vector in a sub-vector normalizer 14 by scaling with the inverse of the corresponding quantized envelope value (gain).

[0021] A bit allocator 16 assigns bits for quantization of different residual sub-vectors based on envelope energies. Due to a limited bit-budget, some of the sub-vectors are not assigned any bits. This is illustrated in Fig. 4, where sub-vectors corresponding to envelope gains below a threshold TH are not assigned any bits. Residual sub-vectors are quantized in a sub-vector quantizer 18 according to the assigned bits. Residual quantization can, for example, be performed with the Factorial Pulse Coding (FPC) scheme [2]. Residual sub-vector quantization indices and envelope quantization indices are then transmitted to the decoder over a multiplexer (MUX) 20.

[0022] At the decoder the received bit stream is de-multiplexed into residual sub-vector quantization indices and envelope quantization indices in a demultiplexer (DEMUX) 22. The residual sub-vector quantization indices are dequantized into residual sub-vectors in a sub-vector dequantizer 24, and the envelope quantization indices are dequantized into envelope gains in an envelope dequantizer 26. A bit allocator 28 uses the envelope gains to control the residual sub-vector dequantization.

[0023] Residual sub-vectors with zero bits assigned have not been coded at the encoder, and are instead noise-filled by a noise filler 30 at the decoder. This is achieved by creating a Virtual Codebook (VC) from coded sub-vectors by concatenating the perceptually relevant coefficients of the decoded spectrum ([1] section 8.4.1). Thus, the VC creates content in the non-coded residual sub-vectors.

[0024] At the decoder, the MDCT vector $\hat{x}(n)$ is then reconstructed by up-scaling residual sub-vectors with corresponding envelope gains in an envelope shaper 32, and transforming the resulting frequency domain vector $\hat{X}(k)$ in an inverse MDCT transformer 34.

[0025] A drawback of the conventional noise-fill scheme described above is that It creates audible distortion in the reconstructed audio signal, when used with the FPC scheme. The main reason is that some of the coded vectors may be too sparse, which creates energy mismatch problems in the noise-filled bands. Additionally some of the coded vectors may contain too much structure (color), which leads to perceptual degradations when the noise-fill is performed at high frequencies.

[0026] The following description will focus on an embodiment of an improved procedure for virtual codebook generation in step H above.

[0027] A coded residual $\hat{X}(k)$, illustrated in Fig. 5, is compressed or quantized according to:

$$Y(k) = \begin{cases} 1 & \text{if} \quad \hat{X}(k) > 0 \\ 0 & \text{if} \quad \hat{X}(k) = 0 \\ -1 & \text{if} \quad \hat{X}(k) < 0 \end{cases} \qquad (1)$$

as illustrated in Fig. 6. This step guarantees that there will be no excessive structure (such as periodicity at high-frequencies) in the noise-filled regions. In addition the specific form of compressed residual $Y(k)$ allows a low complexity in the following steps.

[0028] As an alternative the coded residual $\hat{X}(k)$ may be compressed or quantized according to:

$$Y(k) = \begin{cases} 1 & \text{if} \quad \hat{X}(k) > T \\ 0 & \text{if} \quad -T \le \hat{X}(k) \le T \\ -1 & \text{if} \quad \hat{X}(k) < -T \end{cases} \qquad (2)$$

where T is a small positive number. The value of T may be used to control the amount of compression. This embodiment is also useful for signals that have been coded by an encoder that quantizes symmetrically around 0 but does not include the actual value 0.

[0029] The virtual codebook is built only from "populated" M-dimensional sub-vectors. If a coded residual sub-vector does not fulfill the criterion:

$$\sum_{k=1}^{M}|Y(k)| \geq 2 \qquad (3)$$

it is considered sparse, and is rejected. For example, if the sub-vector has dimension 8 (M=8), equation (3) guarantees that a particular sub-vector will be rejected from the virtual codebook if it has more than 6 zeros. This is illustrated in Fig. 7, where sub-vector SV3 is rejected, since it has 7 zeros. A virtual codebook VC1 is formed by concatenating the remaining or surviving sub-vectors, as illustrated in Fig. 8. Since the length of the sub-vectors is a multiple of M, the criterion (3) may be used also for longer sub-vectors. In this case the parts that do not fulfill the criterion are rejected.

[0030] In general a compressed sub-vector is considered "populated" if it contains more that 20-30% of non-zero components. In the example above with M=8 the criterion is "more than 25% of non-zero components".

[0031] A second virtual codebook VC2 is created from the obtained virtual codebook VC1. This second virtual codebook VC2 is even more "populated" and is used to fill frequencies above 4.8 kHz (other transition frequencies are of course also possible; typically the transition frequency is between 4 and 6 kHz). The second virtual codebook VC2 is formed in accordance with:

$$Z(k) = Y(k) \oplus Y(N-k), \qquad k = 0 \ldots N-1 \qquad (4)$$

where N is the size (total number of coefficients $Y(k)$) of the first virtual codebook VC1, and the combining operation $\oplus$ is defined as:

$$Z(k) = \begin{cases} sign(Y(k)) \times \left(|Y(k)| + |Y(N-k)|\right) & \text{if} \quad Y(k) \neq 0 \\ Y(N-k) & \text{if} \quad Y(k) = 0 \end{cases} \qquad (5)$$

[0032] This combining or merging step is illustrated in Fig. 9A-B. It is noted that the same pair of coefficients $Y(k)$, $Y(N-k)$ is used twice in the merging process, once in the lower half (Fig. 9A) and once in the upper half (Fig. 9B).

[0033] Non-coded sub-vectors may be filled by cyclically stepping through the respective virtual codebook, VC1 or VC2 depending on whether the sub-vector to be filled is below or above the transition frequency, and copying the required number of codebook coefficients to the empty sub-vector. Thus, if the codebooks are short and there are many sub-vectors to be filled, the same coefficients will be reused for filling more than one sub-vector.

[0034] An energy adjustment of the filled sub-vectors is preferably performed on a sub-vector basis. It accounts for the fact that after the spectrum filling the residual sub-vectors may not have the expected unit RMS energy. The adjustment may be performed in accordance with:

$$D(k) = \frac{\alpha}{\sqrt{\frac{1}{M}\sum_{k=1}^{M} Z(k)^2}} Z(k) \qquad (6)$$

where $\alpha \leq 1$, for example $\alpha = 0.8$, is a perceptually optimized attenuation factor. A motivation for the perceptual attenuation is that the noise-fill operation often results in significantly different statistics of the residual vector and it is desirable to attenuate such "inaccurate" regions.

[0035] In a more advanced scheme energy adjustment of a particular sub-vector can be adapted to the type of neighboring sub-vectors: If the neighboring regions are coded at high-bitrate, attenuation of the current sub-vector is more aggressive (alpha goes towards zero). If the neighboring regions are coded at a low-bitrate or noise-filled, attenuation of the current sub-vector is limited (alpha goes towards one). This scheme prevents attenuation of large continuous spectral regions, which might lead to audible loudness loss. At the same time if the spectral region to be attenuated is narrow, even a very strong attenuation will not affect the overall loudness.

[0036] The described technology provides improved noise-filling. Perceptual improvements have been measured by means of

listening tests. These tests indicate that the spectrum fill procedure described above was preferred by listeners in 83% of the tests while the conventional noise fill procedure was preferred in 17% of the tests.

[0037] Fig. 10 is a block diagram illustrating an example of a low frequency virtual codebook generator 60. Residual sub-vectors are forwarded to a sub-vector compressor 42, which is configured to compress actually coded residual sub-vectors (i.e. sub-vectors that have actually been allocated bits for coding), for example in accordance with equation (1). The compressed sub-vectors are forwarded to a sub-vector rejecter 44, which is configured to reject compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion, for example criterion (3). The remaining compressed sub-vectors are collected in a sub-vector collector 46, which is configured to concatenate them to form the low frequency virtual codebook VC 1.

[0038] Fig. 11 is a block diagram illustrating an example of a high frequency virtual codebook generator 70. Residual sub-vectors are forwarded to a sub-vector compressor 42, which is configured to compress actually coded residual sub-vectors (i.e. sub-vectors that have actually been allocated bits for coding), for example in accordance with equation (1). The compressed sub-vectors are forwarded to a sub-vector rejecter 44, which is configured to reject compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion, for example criterion (3). The remaining compressed sub-vectors are collected in a sub-vector collector 46, which is configured to concatenate them to form the low frequency virtual codebook VC1. Thus, up to this point the high frequency virtual codebook generator 70 includes the same elements as the low frequency virtual codebook generator 60. Coefficients from the low frequency virtual codebook VC1 are forwarded to a coefficient combiner 48, which is configured to combine pairs of coefficients to form the high frequency virtual codebook VC2, for example in accordance with equation (5).

[0039] Fig. 12 is a block diagram illustrating an embodiment of a spectrum filler 40. Residual sub-vectors are forwarded to a sub-vector compressor 42, which is configured to compress actually coded residual sub-vectors (i.e. sub-vectors that have actually been allocated bits for coding), for example in accordance with equation (1). The compressed sub-vectors are forwarded to a sub-vector rejecter 44, which is configured to reject compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion, for example criterion (3). The remaining compressed sub-vectors are collected in a sub-vector collector 46, which is configured to concatenate them to form a first (low frequency) virtual codebook VC1. Coefficients from the first virtual codebook VC1 are forwarded to a coefficient combiner 48, which is configured to combine pairs of coefficients to form a second (high frequency) virtual codebook VC2, for example in accordance with equation (5). Thus, up to this point the spectrum filler 40 includes the same elements as the high frequency virtual codebook generator 70. The residual sub-vectors are also forwarded to a sub-vector filler 50, which is configured to fill non-coded residual sub-vectors below a predetermined frequency with coefficients from the first virtual codebook VC1, and to fill non-coded residual sub-vectors above the predetermined frequency with coefficients from the second virtual codebook. In a preferred embodiment the spectrum filler 40 also includes an energy adjuster 52 configured to adjust the energy of filled non-coded residual sub-vectors to obtain a perceptual attenuation, as described above.

[0040] Fig. 13 is a block diagram illustrating an embodiment of a decoder 300 including a spectrum filler 40. The general structure of the decoder 300 is the same as of the decoder in Fig. 1, but with the noise filler 30 replaced by the spectrum filler 40.

[0041] Fig. 14 is a flow chart illustrating low frequency virtual codebook generation. Step S1 compresses actually coded residual sub-vectors, for example in accordance with equation (1). Step S2 rejects compressed residual sub-vectors that are too sparse, i.e. compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion, for example criterion (3). Step S3 concatenates the remaining compressed residual sub-vectors to form the virtual codebook VC1.

[0042] Fig. 15 is a flow chart illustrating high frequency virtual codebook generation. Step S1 compresses actually coded residual sub-vectors, for example in accordance with equation (1). Step S2 rejects compressed residual sub-vectors that are too sparse, i.e. compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion, such as criterion (3). Step S3 concatenates the remaining compressed residual sub-vectors to form a first virtual codebook VC 1. Thus, up to this point the high frequency virtual codebook generation includes the same steps as the low frequency virtual codebook generation. Step S4 combines pairs of coefficients of the first virtual codebook VC1, for example in accordance with equation (5), thereby forming the high frequency virtual codebook VC2.

[0043] Fig. 16 is a flow chart illustrating spectrum filling. Step S1 compresses actually coded residual sub-vectors, for example in accordance with equation (1). Step S2 rejects compressed residual sub-vectors that are too sparse, i.e. compressed residual sub-vectors that do not fulfill a predetermined sparseness criterion, such as criterion (3). Step S3 concatenates the remaining compressed residual sub-vectors to form a first virtual codebook VC 1. Step S4 combines pairs of coefficients of the first virtual codebook VC 1, for example in accordance with equation (5), to form a second virtual codebook VC2. Thus, up to this point the

spectrum filling includes the same steps as the high frequency virtual codebook generation. Step S5 fills non-coded residual sub-vectors below a predetermined frequency with coefficients from the first virtual codebook VC1. Step S6 fills non-coded residual sub-vectors above a predetermined frequency with coefficients from the second virtual codebook VC2. Optional step S7 adjusts the energy of filled non-coded residual sub-vectors to obtain a perceptual attenuation, as described above. Fig. 17 is a block diagram illustrating an example of a low frequency virtual codebook generator 60. This example is based on a processor 110, for example a micro processor, which executes a software component 120 for compressing actually coded residual sub-vectors, a software component 130 for rejecting compressed residual sub-vectors that are too sparse, and a software component 140 for concatenating the remaining compressed residual sub-vectors to form the virtual codebook VC1. These software components are stored in memory 150. The processor 110 communicates with the memory over a system bus. The residual sub-vectors are received by an input/output (I/O) controller 160 controlling an I/O bus, to which the processor 110 and the memory 150 are connected. In this example the residual sub-vectors received by the I/O controller 160 are stored in the memory 150, where they are processed by the software components. Software component 120 may implement the functionality of block 42 in the example described with reference to Fig. 10 above. Software component 130 may implement the functionality of block 44 in the example described with reference to Fig. 10 above. Software component 140 may implement the functionality of block 46 in the example described with reference to Fig. 10 above. The virtual codebook VC1 obtained from software component 140 is outputted from the memory 150 by the I/O controller 160 over the I/O bus or is stored in memory 150.

[0044] Fig. 18 is a block diagram illustrating an example of a high frequency virtual codebook generator 70. This example is based on a processor 110, for example a micro processor, which executes a software component 120 for compressing actually coded residual sub-vectors, a software component 130 for rejecting compressed residual sub-vectors that are too sparse, a software component 140 for concatenating the remaining compressed residual sub-vectors to form low frequency virtual codebook VC1, and a software component 170 for combining coefficient pairs from the codebook VC1 to form the high frequency virtual codebook VC2. These software components are stored in memory 150. The processor 110 communicates with the memory over a system bus. The residual sub-vectors are received by an input/output (I/O) controller 160 controlling an I/O bus, to which the processor 110 and the memory 150 are connected. In this example the residual sub-vectors received by the I/O controller 160 are stored in the memory 150, where they are processed by the software components. Software component 120 may implement the functionality of block 42 in the example described with reference to Fig. 11 above. Software component 130 may implement the functionality of block 44 in the example described with reference to Fig. 11 above. Software component 140 may implement the functionality of block 46 in the example described with reference to Fig. 11 above. Software component 170 may implement the functionality of block 48 in the example described with reference to Fig. 11 above. The virtual codebook VC1 obtained from software component 140 is preferably stored in memory 150 for this purpose. The virtual codebook VC2 obtained from software component 170 is outputted from the memory 150 by the I/O controller 160 over the I/O bus or is stored in memory 150.

[0045] Fig. 19 is a block diagram illustrating an embodiment of a spectrum filler 40. This embodiment is based on a processor 110, for example a micro processor, which executes a software component 180 for generating a low frequency virtual codebook VC1, a software component 190 for generating a high frequency virtual codebook VC2, a software component 200 for filling non-coded residual sub-vectors below a predetermined frequency from the virtual codebook VC1, and a software component 210 for filling non-coded residual sub-vectors above a predetermined frequency from the virtual codebook VC2. These software components are stored in memory 150. The processor 110 communicates with the memory over a system bus. The residual sub-vectors are received by an input/output (I/O) controller 160 controlling an I/O bus, to which the processor 110 and the memory 150 are connected. In this embodiment the residual sub-vectors received by the I/O controller 160 are stored in the memory 150, where they are processed by the software components. Software component 180 may implement the functionality of blocks 42-46 in the embodiment described with reference to Fig. 12 above. Software component 190 may implement the functionality of block 48 in the embodiments described with reference to Fig. 12 above. Software components 200, 210 may implement the functionality of block 50 in the embodiment described with reference to Fig. 12 above. The virtual codebooks VC1, VC2 obtained from software components 180 and 190 are preferably stored in memory 150 for this purpose. The filled residual sub-vectors obtained from software components 200, 201 are outputted from the memory 150 by the I/O controller 160 over the I/O bus or are stored in memory 150.

[0046] The technology described above is intended to be used in an audio decoder, which can be used in a mobile device (e.g. mobile phone, laptop) or a stationary PC. Here the term User Equipment (UE) will be used as a generic name for such devices. An audio decoder with the proposed spectrum fill scheme may be used in real-time communication scenarios (targeting primarily speech) or streaming scenarios (targeting primarily music).

[0047] Fig. 20 illustrates an embodiment of a user equipment in accordance with the present technology. It includes a decoder 300 provided with a spectrum filler 40 in accordance with the present technology. This embodiment illustrates a radio terminal, but other network nodes are also feasible. For example, if voice over IP (Internet Protocol) is used in the network, the user equipment may comprise a computer.

[0048] In the user equipment in Fig. 20 an antenna 302 receives an encoded audio signal. A radio unit 304 transforms this signal into audio parameters, which are forwarded to the decoder 300 for generating a digital audio signal, as described with reference to the various embodiments above. The digital audio signal is then D/A converted and amplified in a unit 306 and finally forwarded to a loudspeaker 308.

[0049] It will be understood by those skilled in the art that various modifications and changes may be made to the present technology without departure from the scope thereof, which is defined by the appended claims.

REFERENCES

[0050]

[1] ITU-T Rec. G.719, "Low-complexity full-band audio coding for high-quality conversational applications," 2008, Sections 8.4.1, 8.4.3.

[2] Mittal, J. Ashley, E. Cruz-Zeno, "Low Complexity Factorial Pulse Coding of MDCT Coefficients using Approximation of Combinatorial Functions," ICASSP 2007

ABBREVIATIONS

[0051]

FPC
: Factorial Pulse Coding

MDCT
: Modified Discrete Cosine Transform

RMS
: Root-Mean-Square

UE
: User Equipment

VC
: Virtual Codebook

REFERENCES CITED IN THE DESCRIPTION

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

Patent documents cited in the description

- US2010241437A1 **[0005]**

Non-patent literature cited in the description

- **MITTAL, J. ASHLEY E. CRUZ-ZENO**Low Complexity Factorial Pulse Coding of MDCT Coefficients using Approximation of Combinatorial FunctionsICASSP, 2007, [0050]

FYLDNING AF IKKE-KODEDE UNDERVEKTORER I TRANSFORMATIONSKODEDE LYDSIGNALER

PATENTKRAV

1. Fremgangsmåde til fyldning af ikke-kodede resterende undervektorer af et transformationskodet lydsignal, hvilken fremgangsmåde indbefatter følgende trin:

komprimering (S1) af aktuelt kodede resterende undervektorer;

afvisning (S2) af komprimerede resterende undervektorer, der ikke opfylder et forhåndsbestemt sjældenhedskriterium;

sammenkædning (S3) af de resterende komprimerede resterende undervektorer for at danne en første virtuel kodebog (VC1);

kombinering af (S4) par af koefficienter fra den første virtuelle kodebog (VC1) for at danne en anden virtuel kodebog (VC2);

fyldning (S5) af ikke-kodede resterende undervektorer under en forhåndsbestemt frekvens med koefficienter fra den første virtuelle kodebog (VC1);

fyldning (S6) af ikke-kodede resterende undervektorer over den forhåndsbestemte frekvens med koefficienter fra den anden virtuelle kodebog,

kendetegnet ved, at komponenter $\hat{X}(k)$ af aktuelt kodede resterende undervektorer komprimeres (S1) i henhold til:

$$Y(k) = \begin{cases} 1 & \text{hvis } \hat{X}(k) > 0 \\ 0 & \text{hvis } \hat{X}(k) = 0 \\ -1 & \text{hvis } \hat{X}(k) < 0 \end{cases}$$

hvor $Y(k)$ er komponenterne af de komprimerede resterende undervektorer.

2. Fremgangsmåde ifølge krav 1, hvor komprimerede resterende undervektorer med mindre end en forhåndsbestemt procentdel af ikke-nul-komponenter afvises (S2).

3. Fremgangsmåde ifølge krav 1 eller 2, hvor par af koefficienter $Y(k)$ fra den første virtuelle kodebog (VC1) kombineres (S4) i henhold til:

$$Z(k) = \begin{cases} sign(Y(k)) \times \left(|Y(k)| + |Y(N-k)| \right) & \text{hvis } Y(k) \neq 0 \\ Y(N-k) & \text{hvis } Y(k) = 0 \end{cases} \qquad k = 0 \ldots N-1$$

hvor N er størrelsen af den første virtuelle kodebog (VC1) og $Z(k)$ er komponenterne af den anden virtuelle kodebog (VC2).

4. Fremgangsmåde ifølge krav 1, 2 eller 3, hvilken fremgangsmåde indbefatter trinnet med justering (S7) af energien af fyldte ikke-kodede resterende undervektorer for at opnå en detekterbar svækkelse.

5. Spektrumfylder (40) til fyldning af ikke-kodede resterende undervektorer af et transformationskodet lydsignal, hvilken spektrumfylder indbefatter:

en undervektorkompressor (42), der er konfigureret til at komprimere aktuelt kodede resterende undervektorer;

en undervektorafviser (44), der er konfigureret til at afvise komprimerede resterende undervektorer, der ikke opfylder et forhåndsbestemt sjældenhedskriterium;

en undervektorkollektor (46), der er konfigureret til at sammenkæde de resterende komprimerede resterende undervektorer for at danne en første virtuel kodebog (VC1);

en koefficientkombinator (48), der er konfigureret til at kombinere par af koefficienter fra den første virtuelle kodebog (VC1) for at danne en anden virtuel kodebog (VC2);

en undervektorfylder (50), der er konfigureret til at fylde ikke-kodede resterende undervektorer under en forhåndsbestemt frekvens med koefficienter fra den første virtuelle kodebog (VC1) og til at fylde ikke-kodede resterende undervektorer over den forhåndsbestemte frekvens med koefficienter fra den anden virtuelle kodebog (VC2), kendetegnet ved, at undervektorkompressoren (42) er konfigureret til at komprimere komponenter $\hat{X}(k)$ af aktuelt kodede resterende undervektorer i henhold til:

$$Y(k) = \begin{cases} 1 & \text{hvis } \hat{X}(k) > 0 \\ 0 & \text{hvis } \hat{X}(k) = 0 \\ -1 & \text{hvis } \hat{X}(k) < 0 \end{cases}$$

hvor $Y(k)$ er komponenterne af de komprimerede resterende undervektorer.

6. Spektrumfylder ifølge krav 5, hvor undervektorafviseren (44) er konfigureret til at afvise komprimerede resterende undervektorer med mindre end en forhåndsbestemt procentdel af ikke-nul-komponenter.

7. Spektrumfylder ifølge krav 5 eller 6, hvor koefficientkombinatoren (48) er konfigureret til at kombinere par af koefficienter $Y(k)$ fra den første virtuelle kodebog (VC1) i henhold til:

$$Z(k) = \begin{cases} sign(Y(k)) \times \left(|Y(k)| + |Y(N-k)| \right) \text{ hvis } & Y(k) \neq 0 \\ & k = 0 \ldots N-1 \\ Y(N-k) \quad \text{hvis } & Y(k) = 0 \end{cases}$$

hvor N er størrelsen af den første virtuelle kodebog (VC1) og $Z(k)$ er komponenterne af den anden virtuelle kodebog (VC2).

8. Spektrumfylder ifølge krav 5, 6 eller 7, hvilken spektrumfylder indbefatter en energitilpasser (52), der er konfigureret til at justere energien for fyldte ikke-kodede resterende undervektorer for at opnå en detekterbar svækkelse.

9. Dekoder (300) indbefattende en spektrumfylder (40) ifølge et hvilket som helst af de foregående krav 5-8.

10. Brugerudstyr (UE) indbefattende en dekoder ifølge krav 9.

DRAWINGS

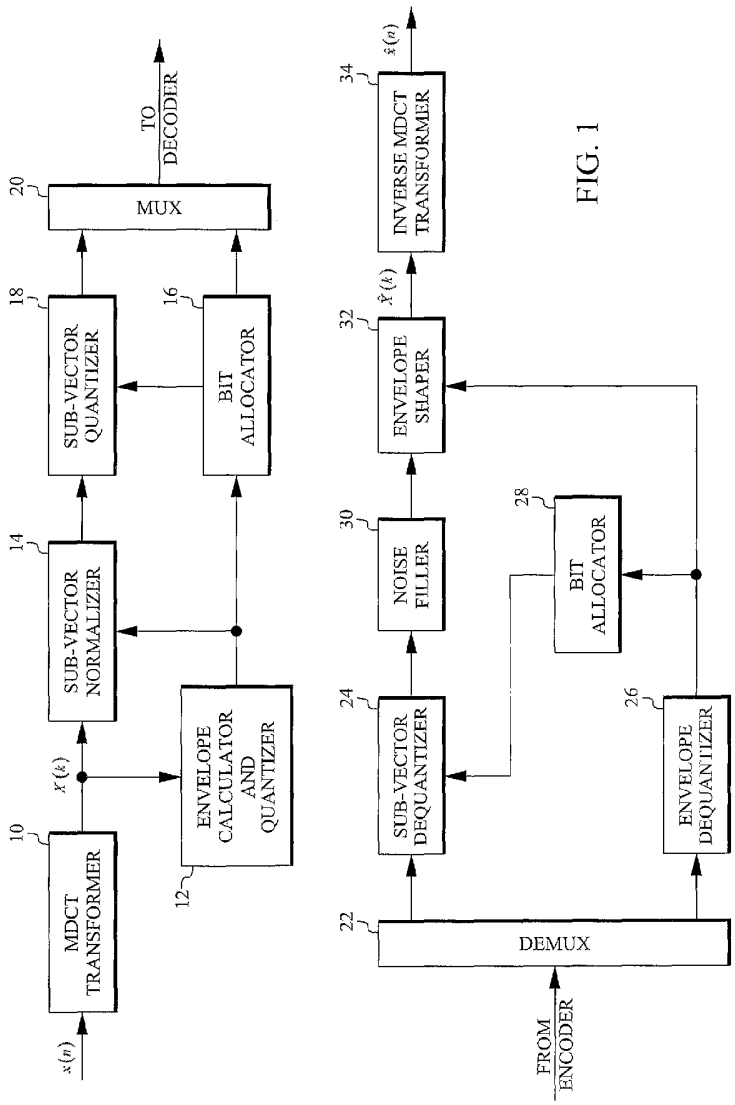


FIG. 1

Fig. 2

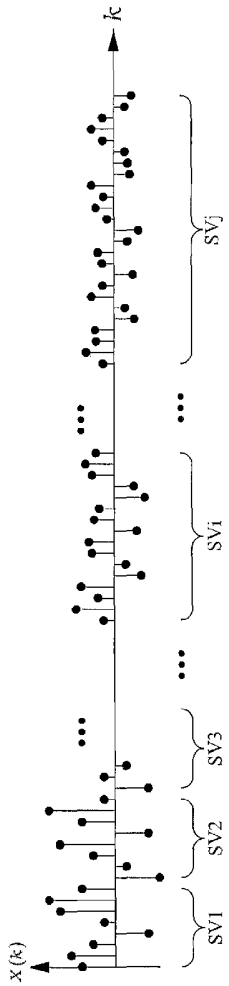


Fig. 3

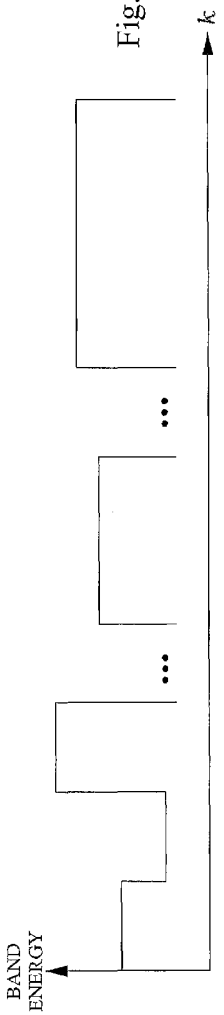
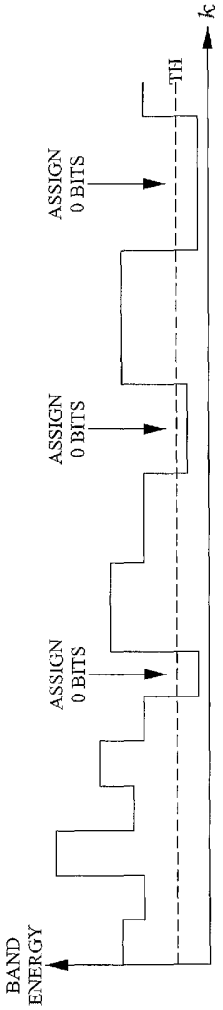


Fig. 4



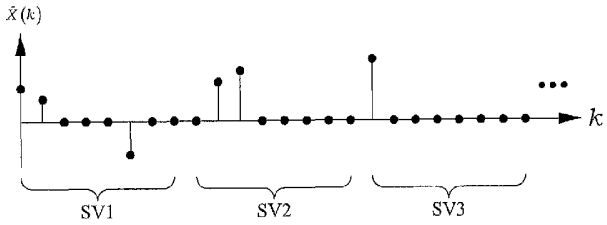


Fig. 5

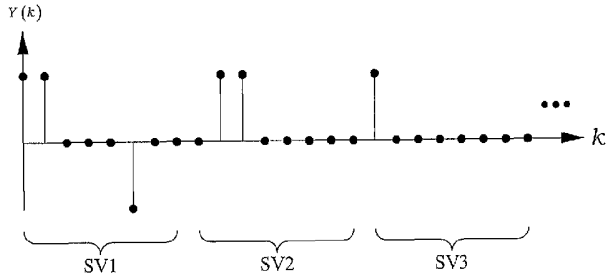


Fig. 6

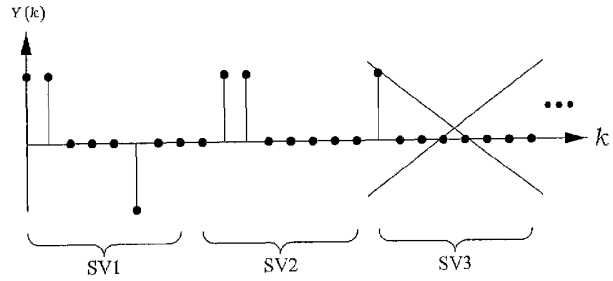


Fig. 7

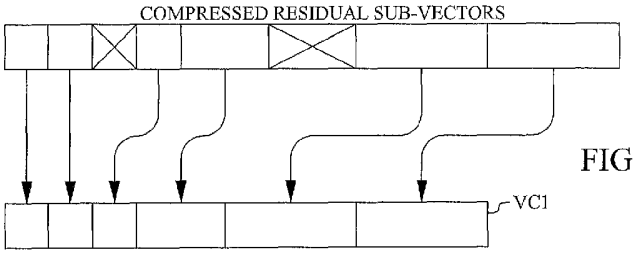


FIG. 8

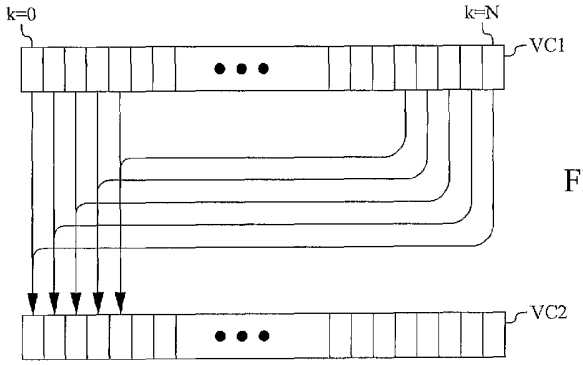


FIG. 9A

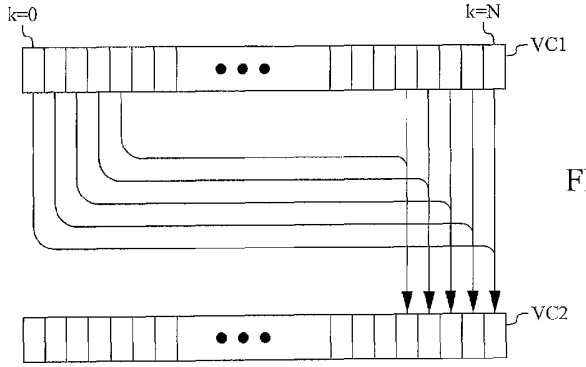


FIG. 9B

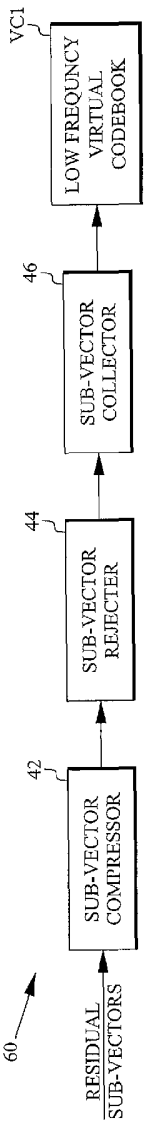

60
RESIDUAL SUB-VECTORS
42
SUB-VECTOR COMPRESSOR
44
SUB-VECTOR REJECTER
46
SUB-VECTOR COLLECTOR
VC1
LOW FREQUNCY VIRTUAL CODEBOOK

FIG. 10

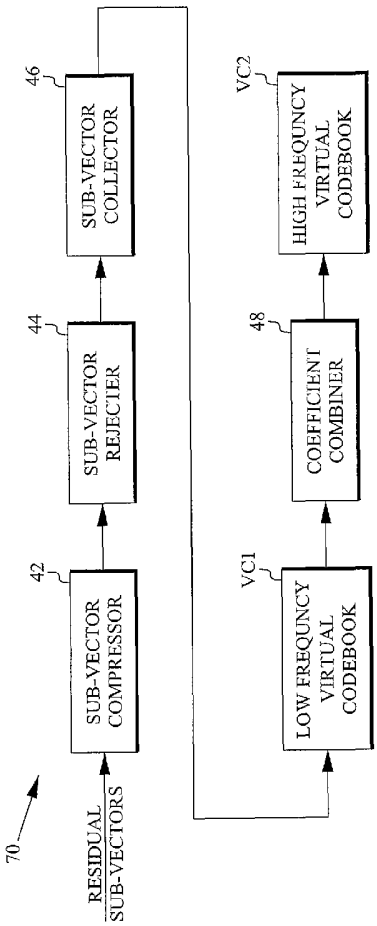

70
RESIDUAL SUB-VECTORS
42
SUB-VECTOR COMPRESSOR
44
SUB-VECTOR REJECTER
46
SUB-VECTOR COLLECTOR
VC1
LOW FREQUNCY VIRTUAL CODEBOOK
48
COEFFICIENT COMBINER
VC2
HIGH FREQUNCY VIRTUAL CODEBOOK

FIG. 11

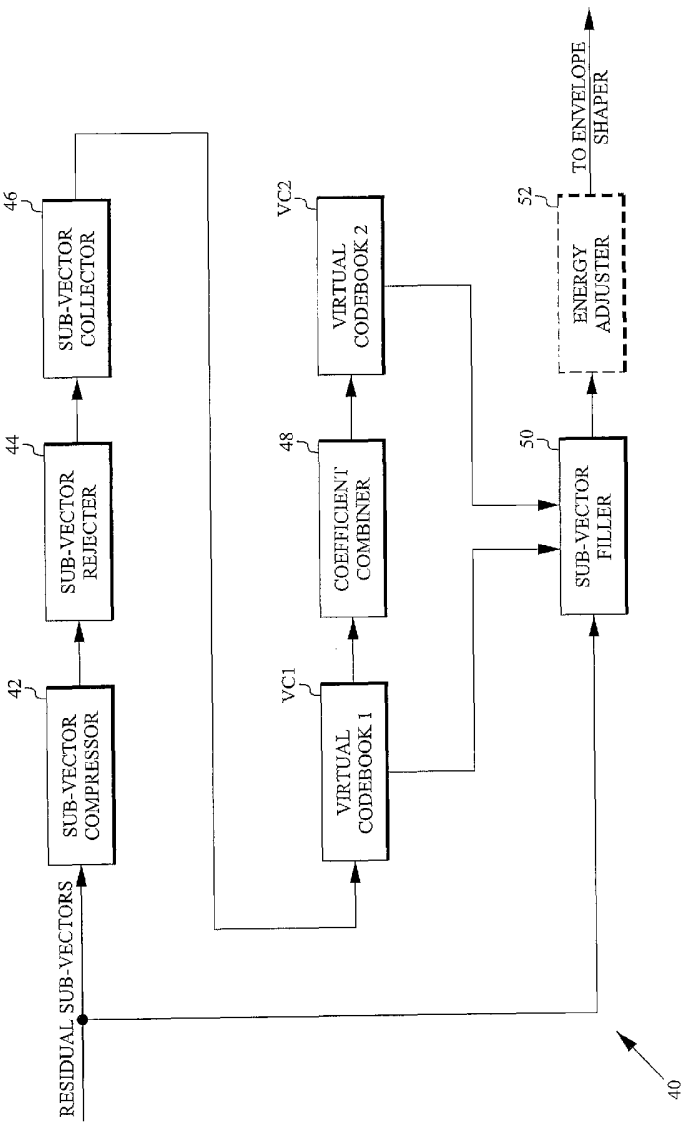

RESIDUAL SUB-VECTORS
SUB-VECTOR COMPRESSOR
42
SUB-VECTOR REJECTER
44
SUB-VECTOR COLLECTOR
46
VIRTUAL CODEBOOK 1
VC1
COEFFICIENT COMBINER
48
VIRTUAL CODEBOOK 2
VC2
SUB-VECTOR FILLER
50
ENERGY ADJUSTER
52
TO ENVELOPE SHAPER
40

FIG. 12

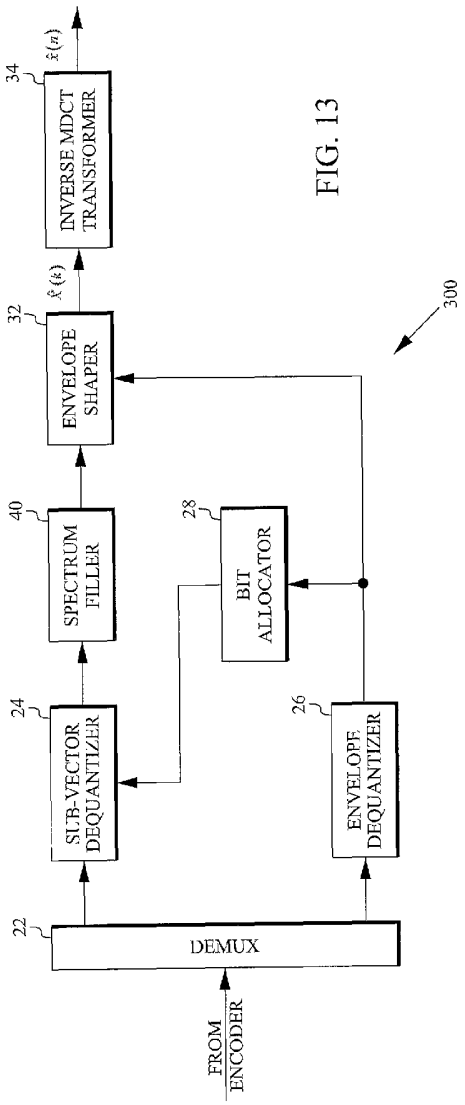

FROM ENCODER
DEMUX
22
SUB-VECTOR DEQUANTIZER
24
ENVELOPE DEQUANTIZER
26
BIT ALLOCATOR
28
SPECTRUM FILLER
40
ENVELOPE SHAPER
32
INVERSE MDCT TRANSFORMER
34
300

FIG. 13

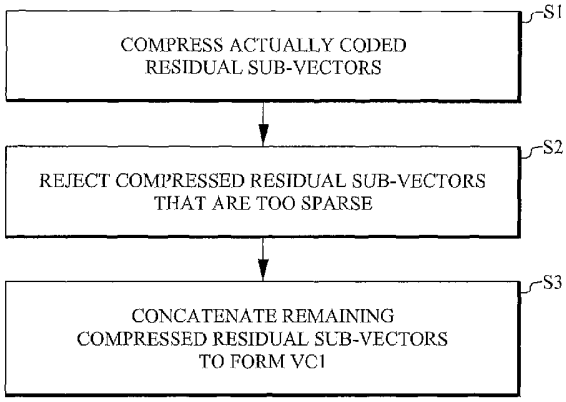

S1
COMPRESS ACTUALLY CODED RESIDUAL SUB-VECTORS
S2
REJECT COMPRESSED RESIDUAL SUB-VECTORS THAT ARE TOO SPARSE
S3
CONCATENATE REMAINING COMPRESSED RESIDUAL SUB-VECTORS TO FORM VC1

FIG. 14

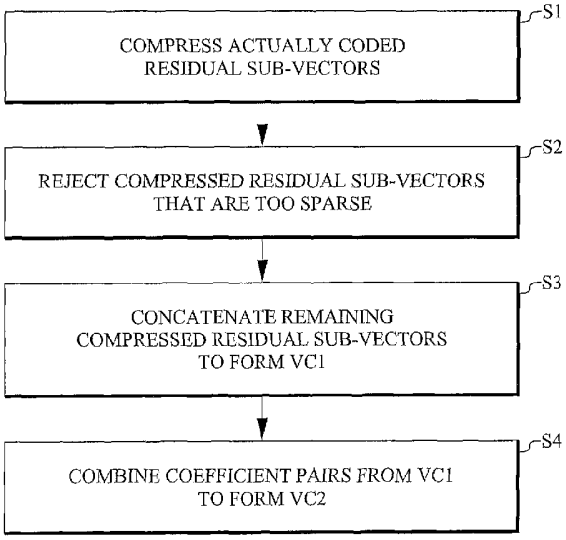

S1
COMPRESS ACTUALLY CODED RESIDUAL SUB-VECTORS
S2
REJECT COMPRESSED RESIDUAL SUB-VECTORS THAT ARE TOO SPARSE
S3
CONCATENATE REMAINING COMPRESSED RESIDUAL SUB-VECTORS TO FORM VC1
S4
COMBINE COEFFICIENT PAIRS FROM VC1 TO FORM VC2

FIG. 15

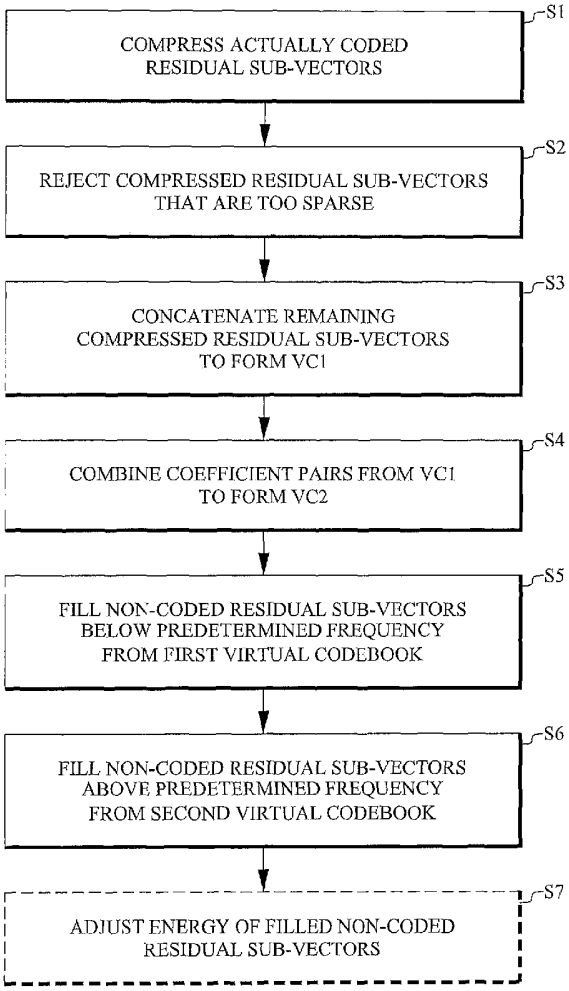

S1
COMPRESS ACTUALLY CODED RESIDUAL SUB-VECTORS
S2
REJECT COMPRESSED RESIDUAL SUB-VECTORS THAT ARE TOO SPARSE
S3
CONCATENATE REMAINING COMPRESSED RESIDUAL SUB-VECTORS TO FORM VC1
S4
COMBINE COEFFICIENT PAIRS FROM VC1 TO FORM VC2
S5
FILL NON-CODED RESIDUAL SUB-VECTORS BELOW PREDETERMINED FREQUENCY FROM FIRST VIRTUAL CODEBOOK
S6
FILL NON-CODED RESIDUAL SUB-VECTORS ABOVE PREDETERMINED FREQUENCY FROM SECOND VIRTUAL CODEBOOK
S7
ADJUST ENERGY OF FILLED NON-CODED RESIDUAL SUB-VECTORS

FIG. 16

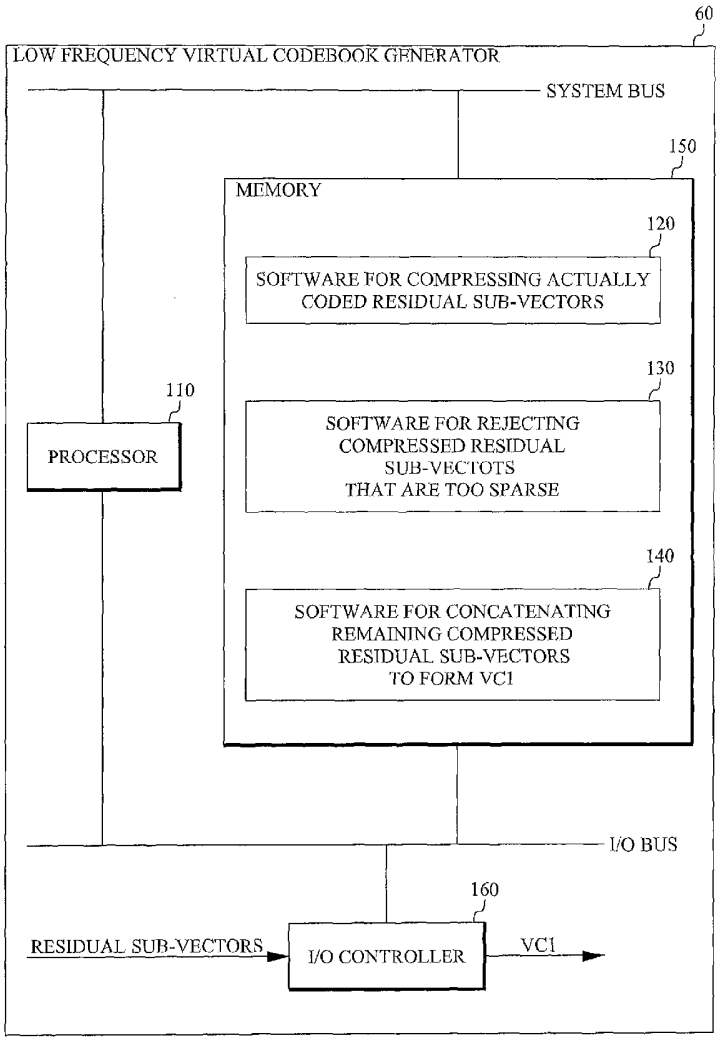

60
LOW FREQUENCY VIRTUAL CODEBOOK GENERATOR
SYSTEM BUS
150
MEMORY
120
SOFTWARE FOR COMPRESSING ACTUALLY CODED RESIDUAL SUB-VECTORS
130
SOFTWARE FOR REJECTING COMPRESSED RESIDUAL SUB-VECTOTS THAT ARE TOO SPARSE
140
SOFTWARE FOR CONCATENATING REMAINING COMPRESSED RESIDUAL SUB-VECTORS TO FORM VC1
110
PROCESSOR
I/O BUS
160
RESIDUAL SUB-VECTORS
I/O CONTROLLER
VC1

FIG. 17

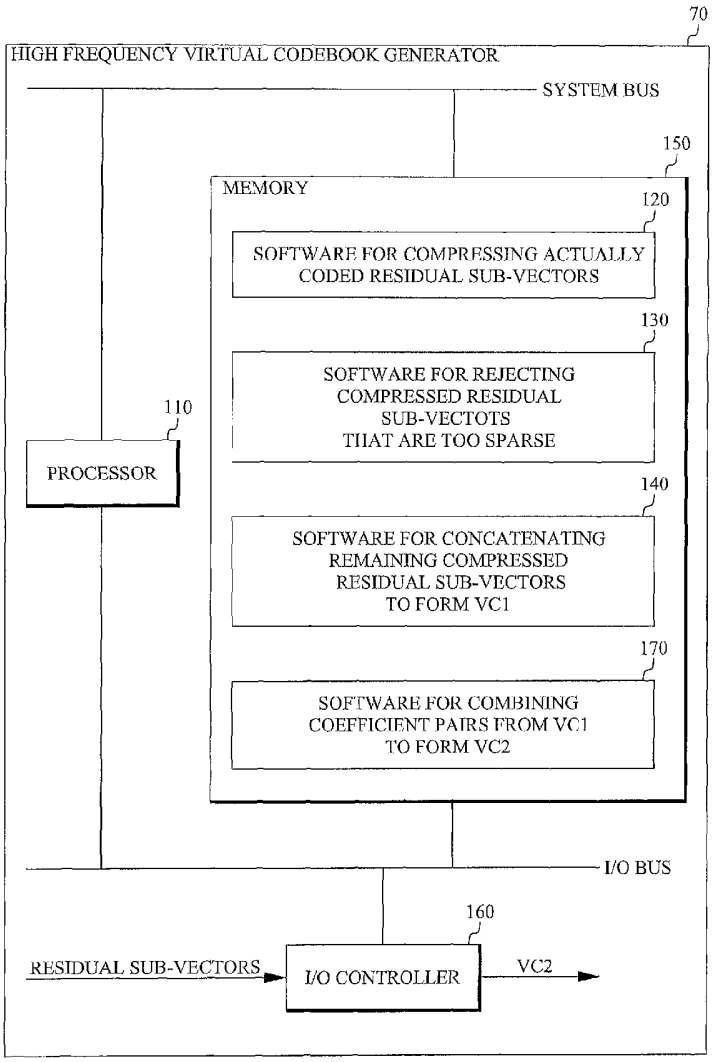

70
HIGH FREQUENCY VIRTUAL CODEBOOK GENERATOR
SYSTEM BUS
150
MEMORY
120
SOFTWARE FOR COMPRESSING ACTUALLY CODED RESIDUAL SUB-VECTORS
130
SOFTWARE FOR REJECTING COMPRESSED RESIDUAL SUB-VECTOTS TIIAT ARE TOO SPARSE
110
PROCESSOR
140
SOFTWARE FOR CONCATENATING REMAINING COMPRESSED RESIDUAL SUB-VECTORS TO FORM VC1
170
SOFTWARE FOR COMBINING COEFFICIENT PAIRS FROM VC1 TO FORM VC2
I/O BUS
160
RESIDUAL SUB-VECTORS
I/O CONTROLLER
VC2

FIG. 18

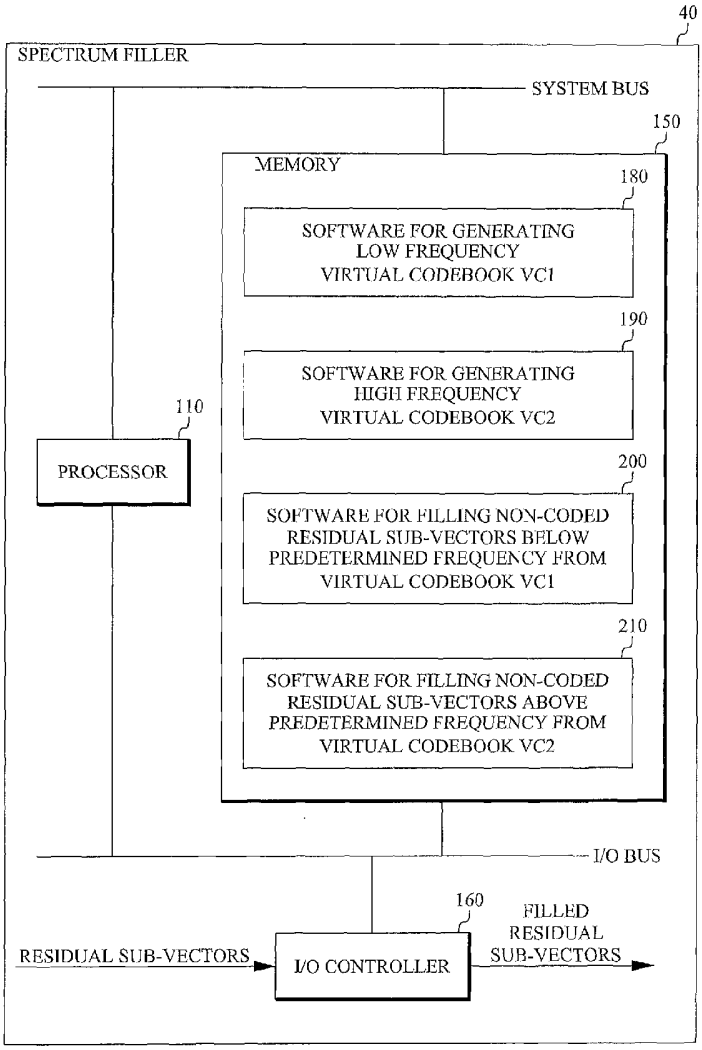

40
SPECTRUM FILLER
SYSTEM BUS
150
MEMORY
180
SOFTWARE FOR GENERATING LOW FREQUENCY VIRTUAL CODEBOOK VC1
190
SOFTWARE FOR GENERATING HIGH FREQUENCY VIRTUAL CODEBOOK VC2
110
PROCESSOR
200
SOFTWARE FOR FILLING NON-CODED RESIDUAL SUB-VECTORS BELOW PREDETERMINED FREQUENCY FROM VIRTUAL CODEBOOK VC1
210
SOFTWARE FOR FILLING NON-CODED RESIDUAL SUB-VECTORS ABOVE PREDETERMINED FREQUENCY FROM VIRTUAL CODEBOOK VC2
I/O BUS
160
FILLED RESIDUAL SUB-VECTORS
RESIDUAL SUB-VECTORS
I/O CONTROLLER

FIG. 19

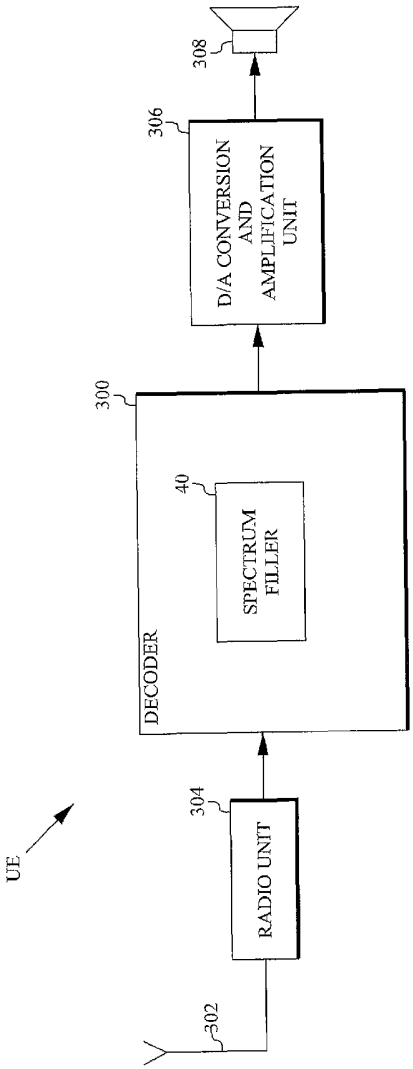

UE
302
304
RADIO UNIT
300
DECODER
40
SPECTRUM FILLER
306
D/A CONVERSION AND AMPLIFICATION UNIT
308

FIG. 20