

**Encoder, Decoder and Methods for Encoding and Decoding Data
Segments representing a time-domain Data Stream**

Description

5

The present invention is in the field of coding, where different characteristics of data to be encoded are utilized for coding rates, as for example in video and audio coding.

10

State of the art coding strategies can make use of characteristics of a data stream to be encoded. For example, in audio coding, perception models are used in order to compress source data almost without decreasing the noticeable quality and degradation when replayed. Modern perceptual audio coding schemes, such as for example, MPEG-2/4 AAC (MPEG = Moving Pictures Expert Group, AAC = Advanced Audio Coding), cf. Generic Coding of Moving Pictures and Associated Audio: Advanced Audio Coding, International Standard 13818-7, ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group, 1997, may use filter banks, such as for example the Modified Discrete Cosine Transform (MDCT), for representing the audio signal in the frequency domain.

25

In the frequency domain quantization of frequency coefficients can be carried out, according to a perceptual model. Such coders can provide excellent perceptual audio quality for general types of audio signals as, for example, music. On the other hand, modern speech coders, such as, for example, ACELP (ACELP = Algebraic Code Excited Linear Prediction), use a predictive approach, and in this way may represent the audio/speech signal in the time domain. Such speech coders can model the characteristics of the human speech production process, i.e. the human vocal tract and, consequently, achieve excellent performance for speech signals at low bit rates. Conversely, perceptual audio coders do not achieve the level of performance offered by

30
35

speech coders for speech signals coded at low bit rates, and using speech coders to represent general audio signals/music results in significant quality impairments.

- 5 Conventional concepts provide a layered combination in which always all partial coders are active, i.e. time-domain and frequency-domain encoders, and the final output signal is calculated by combining the contributions of the partial coders for a given processed time frame. A popular
 10 example of layered coding are MPEG-4 scalable speech/audio coding with a speech coder as the base layer and a filterbank-based enhancement layer, cf. Bernhard Grill, Karlheinz Brandenburg, "A Two-or Three-Stage Bit-Rate Scalable Audio Coding System", Preprint Number 4132, 99th
 15 Convention of the AES (September 1995).

Conventional frequency-domain encoders can make use of MDCT filterbanks. The MDCT has become a dominant filterbank for conventional perceptual audio coders because of its
 20 advantageous properties. For example, it can provide a smooth cross-fade between processing blocks. Even if a signal in each processing block is altered differently, for example due to quantization of spectral coefficients, no blocking artifacts due to abrupt transitions from block to
 25 block occur because of the windowed overlap/add operations. The MDCT uses the concept of time-domain aliasing cancellation (TDAC).

The MDCT is a Fourier-related transform based on the type-
 30 IV discrete cosine transform, with the additional property of being lapped. It is designed to be performed in consecutive blocks of a larger data set, where subsequent blocks are overlapped so that the last half of one block coincides with the first half of the next block. This
 35 overlapping, in addition to an energy-compaction quality of the DCT, makes the MDCT especially attractive for signal compression applications, since it helps to avoid said artifacts stemming from the block boundaries. As a lapped

transform, the MDCT is a bit unusual compared to other Fourier-related transforms in that it has half as many outputs as inputs, instead of the same number. In particular, $2N$ real numbers are transformed into N real numbers, where N is a positive integer.

The inverse MDCT is also known as IMDCT. Because there are different numbers of inputs and outputs, at first glance it might seem that the MDCT should not be invertible. However, perfect invertibility is achieved by adding the overlap IDMTs of subsequent overlapping blocks, causing the errors to cancel and the original data to be retrieved, i.e. achieving TDAC.

Therewith, the number of spectral values at the output of a filterbank is equal to the number of time-domain input values at its input which is also referred to as critical sampling.

An MDCT filterbank provides a high-frequency selectivity and enables a high coding gain. The properties of overlapping of blocks and critical sampling can be achieved by utilizing the technique of time-domain aliasing cancellation, cf. J. Princen, A. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation", IEEE Trans. ASSP, ASSP-34(5):1153-1161, 1986. Fig. 4 illustrates these effects of an MDCT. Fig. 4 shows an MDCT input signal, in terms of an impulse along a time axis 400 at the top. The input signal 400 is then transformed by two consecutive windowing and MDCT blocks, where the windows 410 are illustrated underneath the input signal 400 in Fig. 4. The back transformed individual windowed signals are displayed in Fig. 4 by the time lines 420 and 425.

After the inverse MDCT, the first block produces an aliasing component with positive sign 420, the second block produces an aliasing component with the same magnitude and

a negative sign 425. The aliasing components cancel each other after addition of the two output signals 420 and 425 as shown in the final output 430 at the bottom of Fig. 4.

- 5 In "Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec", 3GPP TS 26.290V6.3.0, 2005-06, Technical Specification the AMR-WB+ (AMR-WB = Adaptive Multi-Rate Wideband) codec is specified. According to section 5.2, the encoding algorithm at the core of the AMR-WB+ codec is
- 10 based on a hybrid ACELP/TCX (TCX = Transform coded Excitation) model. For every block of an input signal the encoder decides, either in an open loop or a closed loop mode which encoding model, i.e. ACELP or TCX, is best. The ACELP model is a time-domain, predictive encoder, best
- 15 suited for speech and transient signals. The AMR-WB encoder is used in ACELP modes. Alternatively, the TCX model is a transform based encoder, and is more appropriate for typical music samples.
- 20 Specifically, the AMR-WB+ uses a discrete Fourier transform (DFT) for the transform coding mode TCX. In order to allow a smooth transition between adjacent blocks, a windowing and overlap is used. This windowing and overlap is necessary both for transitions between different coding
- 25 modes (TCX/ACELP) and for consecutive TCX frames. Thus, the DFT together with the windowing and overlap represents a filterbank that is not critically sampled. The filterbank produces more frequency values than the number of new input samples, cf. Fig. 4 in 3GPP TS 26.290V6.3.0 (3GPP = Third
- 30 Generation Partnership Project, TS = Technical Specification). Each TCX frame utilizes an overlap of 1/8 of the frame length which equals the number of new input samples. Consequently, the corresponding length of the DFT is 9/8 of the frame length.
- 35 Considering the non-critically sampled DFT filterbank in the TCX, i.e. the number of spectral values at the output of the filterbank is larger than the number of time-domain

input values at its input, this frequency domain coding mode is different from audio codecs such as AAC (AAC = Advanced Audio Coding) which utilizes an MDCT, a critically sampled lapped transform.

The Dolby E codec is described in Fielder, Louis D.; Todd, Craig C., "The Design of a Video Friendly Audio Coding System for Distributing Applications", Paper Number 17-008, The AES 17th International Conference: High-Quality Audio Coding (August 1999) and Fielder, Louis D.; Davidson, Grant A., "Audio Coding Tools for Digital Television Distribution", Preprint Number 5104, 108th Convention of the AES (January 2000). The Dolby E codec utilizes the MDCT filterbank. In the design of this coding, special focus was put on the possibility to perform editing in the coding domain. To achieve this, special alias-free windows are used. At the boundaries of these windows a smooth-cross fade or splicing of different signal portions is possible. In the above-referenced documents it is, for example, outlined, cf. section 3 of "The Design of a Video Friendly Audio Coding System for Distribution Applications", that this would not be possible by simply using the usual MDCT windows which introduce time-domain aliasing. However, it is also described that the removal of aliasing comes at the cost of an increased number of transform coefficients, indicating that the resulting filterbank does not have the property of critical sampling anymore.

It is the object of the present invention to provide a more efficient concept for encoding and decoding data segments.

The object is achieved by an apparatus for decoding according to claim 1, a method for decoding according to claim 24, an apparatus for generating an encoded data stream according to claim 26 and a method for generating an encoded data stream according to claim 38.

The present invention is based on the finding that a more efficient encoding and decoding concept can be utilized by using combined time-domain and frequency-domain encoders, respectively decoders. The problem of time aliasing can be efficiently combat by transforming time-domain data to the frequency-domain in the decoder and by combining the resulting transformed frequency-domain data with the decoded frequency-domain data received. Overheads can be reduced by adapting overlapping regions of overlap windows being applied to data segments to coding domain changes. Using windows with smaller overlapping regions can be beneficial when using time-domain encoding, respectively when switching from or to time-domain encoding.

Embodiments can provide a universal audio encoding and decoding concept that achieves improved performance for both types of input signals, such as speech signals and music signals. Embodiments can take advantage by combining multiple coding approaches, e.g. time-domain and frequency-domain coding concepts. Embodiments can efficiently combine filterbank based and time-domain based coding concepts into a single scheme. Embodiments may result in a combined codec which can, for example, be able to switch between an audio codec for music-like audio content and a speech codec for speech-like content. Embodiments may utilize this switching frequently, especially for mixed content.

Embodiments of the present invention may provide the advantage that no switching artifacts occur. In embodiments the amount of additional transmit data, or additionally coded samples, for a switching process can be minimized in order to avoid a reduced efficiency during this phase of operation. Therewith the concept of switched combination of partial coders is different from that of the layered combination in which always all partial coders are active.

In the following embodiments of the present invention will be described in detail using the accompanying Figures, in which

5 Fig. 1a shows an embodiment of an apparatus for decoding;

Fig. 1b shows another embodiment of an apparatus for decoding;

10 Fig. 1c shows another embodiment of an apparatus for decoding;

Fig. 1d shows another embodiment of an apparatus for decoding;

15

Fig. 1e shows another embodiment of an apparatus for decoding;

20 Fig. 1f shows another embodiment of an apparatus for decoding;

Fig. 2a shows an embodiment of an apparatus for encoding;

25 Fig. 2b shows another embodiment of an apparatus for encoding;

Fig. 2c shows another embodiment of an apparatus for encoding;

30 Fig. 3a illustrates overlapping regions when switching between frequency-domain and time-domain coding for the duration of one window;

35 Fig. 3b illustrates the overlapping regions when switching between frequency-domain coding and time-domain coding for a duration of two windows;

Fig. 3c illustrates multiple windows with different overlapping regions;

Fig. 3d illustrates the utilization of windows with
5 different overlapping regions in an embodiment; and

Fig. 4 illustrates time-domain aliasing cancellation when using MDCT.

10 Fig. 1a shows an apparatus 100 for decoding data segments representing a time-domain data stream, a data segment being encoded in a time domain or in a frequency domain, a data segment being encoded in the frequency domain having
15 successive blocks of data representing successive and overlapping blocks of time-domain data samples. This data stream could, for example, correspond to an audio stream, wherein some of the data blocks are encoded in the time domain and other ones are encoded in the frequency domain. Data blocks or segments which have been encoded in the
20 frequency domain, may represent time-domain data samples of overlapping data blocks.

The apparatus 100 comprises a time-domain decoder 110 for decoding a data segment being encoded in the time domain.
25 Furthermore, the apparatus 100 comprises a processor 120 for processing the data segment being encoded in the frequency domain and output data of the time-domain decoder 110 to obtain overlapping time-domain data blocks. Moreover, the apparatus 100 comprises an overlap/add-
30 combiner 130 for combining the overlapping time-domain data blocks to obtain the decoded data segments of the time-domain data stream.

Fig. 1b shows another embodiment of the apparatus 100. In
35 embodiments the processor 120 may comprise a frequency-domain decoder 122 for decoding data segments being encoded in the frequency domain to obtain frequency-domain data segments. Moreover, in embodiments the processor 120 may

comprise a time-domain to frequency-domain converter 124 for converting the output data of the time-domain decoder 110 to obtain converted frequency-domain data segments.

5 Furthermore, in embodiments the processor 120 may comprise a frequency-domain combiner 126 for combining the frequency-domain segments and the converted frequency-domain data segments to obtain a frequency-domain data stream. The processor 120 may further comprise a frequency-
10 domain to time-domain converter 128 for converting the frequency-domain data stream to overlapping time-domain data blocks which can then be combined by the overlap/add-combiner 130.

15 Embodiments may utilize an MDCT filterbank, as for example, used in MPEG-4 AAC, without any modifications, especially without giving up the property of critical sampling. Embodiments may provide optimum coding efficiency. Embodiments may achieve a smooth transition to a time-
20 domain codec compatible with the established MDCT windows while introducing no additional switching artifacts and only a minimal overhead.

Embodiments may keep the time-domain aliasing in the
25 filterbank and intentionally introduce a corresponding time-domain aliasing into the signal portions coded by the time-domain codec. Thus, resulting components of the time-domain aliasing can cancel each other out in the same way as they do for two consecutive frames of the MDCT spectra.

30 Fig. 1c illustrates another embodiment of an apparatus 100. According to Fig. 1c the frequency-domain decoder 122 can comprise a re-quantization stage 122a. Moreover, the time-domain to frequency-domain converter 124 can comprise a
35 cosine modulated filterbank, an extended lapped transform, a low delay filterbank or a polyphase filterbank. The embodiment shown in Fig. 1c illustrates that the time-

domain to frequency-domain converter 124 can comprise an MDCT 124a.

Furthermore, Fig. 1c depicts that the frequency-domain
5 combiner 126 may comprise an adder 126a. As shown in Fig. 1c, the frequency-domain to time-domain converter 128 can comprise a cosine modulated filterbank, respectively an inverse MDCT 128a. The data stream comprising time-domain encoded and frequency-domain encoded data segment may be
10 generated by an encoder which will be further detailed below. The switching between frequency-domain encoding and time-domain encoding can be achieved by encoding some portions of the input signal with a frequency-domain encoder and some input signal portions with a time-domain
15 encoder. The embodiment of the apparatus 100 depicted in Fig. 1c illustrates the principle structure of a corresponding apparatus 100 for decoding. In other embodiments the re-quantization 122a and the inverse modified discrete cosine transform 128a can represent a
20 frequency-domain decoder.

As indicated in Fig. 1c for signal portions where the time-domain decoder 110 takes over, the time-domain output of the time-domain decoder 110 can be transformed by the
25 forward MDCT 124a. The time-domain decoder may utilize a prediction filter to decode the time-domain encoded data. Some overlap in the input of the MDCT 124a and thus some overhead may be introduced here. In the following embodiments will be described which reduce or minimize this
30 overhead.

In principle, the embodiment shown in Fig. 1c also comprises an operation mode where both codecs can operate in parallel. In embodiments the processor 120 can be
35 adapted for processing a data segment being encoded in parallel in the time domain and in the frequency domain. In this way the signal can partially be coded in the frequency domain and partially in the time domain, similar to a

layered coding approach. The resulting signals are then added up in the frequency domain, compare the frequency-domain combiner 126a. Nevertheless, embodiments may carry out a mode of operation which is to switch exclusively
5 between the two codecs and only have a preferably minimum number of samples where both codecs are active in order to obtain best possible efficiency.

In Fig. 1c, the output of the time-domain decoder 110 is
10 transformed by the MDCT 124a, followed by the IMDCT 128a. In another embodiment, these two steps may be advantageously combined into a single step in order to reduce complexity. Fig. 1d illustrates an embodiment of an apparatus 100 illustrating this approach. The apparatus 100
15 shown in Fig. 1d illustrates that the processor 120 may comprise a calculator 129 for calculating overlapping time-domain data blocks based on the output data of the time-domain decoder 110. The processor 120 or the calculator 129 can be adapted for reproducing a property respectively an
20 overlapping property of the frequency-domain to time-domain converter 128 based on the output data of the time-domain decoder 110, i.e. the processor 120 or calculator 129 may reproduce an overlapping characteristic of time-domain data blocks similar to an overlapping characteristic produced by
25 the frequency-domain to time-domain converter 128. Moreover, the processor 120 or calculator 129 can be adapted for reproducing time-domain aliasing similar to time-domain aliasing introduced by the frequency-domain to time-domain converter 128 based on the output data of the
30 time-domain decoder 110.

The frequency-domain to time-domain converter 128 can then be adapted for converting the frequency-domain data segments provided by the frequency-domain decoder 122 to
35 overlapping time-domain data blocks. The overlap/add-combiner 130 can be adapted for combining data blocks provided by the frequency-domain to time-domain converter

128 and the calculator 129 to obtain the decoded data segments of the time-domain data stream.

The calculator 129 may comprise a time-domain aliasing stage 129a as it is illustrated in the embodiment shown in Fig. 1e. The time-domain aliasing stage 129a can be adapted for time-aliasing output data of the time-domain decoder to obtain the overlapping time-domain data blocks.

For the time-domain encoded data a combination of the MDCT and the IMDCT can make the process in embodiments much simpler in both structure and computational complexity as only the process of time-domain aliasing (TDA) remains in embodiments. This efficient process can be based on a number of observations. The windowed MDCT of the input segments of $2N$ samples can be decomposed into three steps.

First, the input signal is multiplied by an analysis window.

Second, the result is then folded down from $2N$ samples to N samples. For the MDCT, this process implies that the first quarter of the samples is combined, i.e. subtracted, in time-reversed order with the second quarter of the samples, and that the fourth quarter of the samples is combined, i.e. added, with the third quarter of the samples in time-reversed order. The result is the time-aliased, down-sampled signal in the modified second and third quarter of the signal, comprising N samples.

Third, the down-sampled signal is then transformed using an orthogonal DCT-like transform mapping N input to N output samples to form the final MDCT output.

The windowed IMDCT reconstruction of an input sequence of N spectral samples can likewise be decomposed into three steps.

First, the input sequence of N spectral samples is transformed using an orthogonal inverse DCT-like transform mapping N input to N output samples.

5 Second, the results unfolded from N to $2N$ samples by writing the inverse DCT transformed values into the second and third quarter of a $2N$ samples output buffer, filling the first quarter with the time-reversed and inverted version of the second quarter, and the fourth quarter with
10 a time-reverse version of the third quarter, respectively.

Third, the resulting $2N$ samples are multiplied with the synthesis window to form the windowed IMDCT output.

15 Thus, a concatenation of the windowed MDCT and the windowed IMDCT may be efficiently carried out in embodiments by the sequence of the first and second steps of the windowed MDCT and the second and third steps of the windowed IMDCT. The third step of the MDCT and the first step of the IMDCT can
20 be omitted entirely in embodiments because they are inverse operations with respect to each other and thus cancel out. The remaining steps can be carried out in the time domain only, and thus embodiments using this approach can be substantially low in computational complexity.

25 For one block of MDCT and consecutive IMDCT, the second and third step of the MDCT and the second and third step of the IMDCT can be written as a multiplication with the following sparse $2N \times 2N$ matrix.

30

$$\begin{bmatrix} 1 & & & -1 & 0 & \dots & \dots & 0 \\ & \ddots & & & \vdots & & & \vdots \\ & & \ddots & & \vdots & & & \vdots \\ -1 & & & 1 & 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & 1 & & & 1 \\ \vdots & & & \vdots & & \ddots & \ddots & \\ \vdots & & & \vdots & & \ddots & \ddots & \\ 0 & \dots & \dots & 0 & 1 & & & 1 \end{bmatrix}$$

In other words, the calculator 129 can be adapted for
 5 segmenting the output of the time-domain decoder 110 in
 calculator segments comprising $2N$ sequential samples,
 applying weights to the $2N$ samples according to an analysis
 windowing function, subtracting the first $N/2$ samples in
 reversed order from the second $N/2$ samples, and the last
 10 $N/2$ samples in reversed order to the third $N/2$ samples,
 inverting the second and third $N/2$ samples, replacing the
 first $N/2$ samples with the time-reversed and inverted
 version of the second $N/2$ samples, replacing the fourth $N/2$
 samples with the time reversed version of the third $N/2$
 15 samples, and applying weights to the $2N$ samples according
 to a synthesis windowing function.

In other embodiments the overlap/add-combiner 130 can be
 adapted for applying weights according to a synthesis
 20 windowing function to overlapping time-domain data blocks
 provided by the frequency-domain to time-domain converter
 128. Furthermore, the overlap/add-combiner 130 can be
 adapted for applying weights according to a synthesis
 windowing function being adapted to the size of an
 25 overlapping region of consecutive overlapping time-domain
 data blocks.

The calculator 129 may be adapted for applying weights to
 the $2N$ samples according to an analysis windowing function
 30 being adapted to the size of an overlapping region of

consecutive overlapping time-domain data blocks and the calculator may be further adapted for applying weights to the $2N$ samples according to a synthesis window function being adapted to the size of the overlapping region.

5

In embodiments the size of an overlapping region of two consecutive time-domain data blocks which are encoded in the frequency-domain can be larger than the size of an overlapping of two consecutive time-domain data blocks of which one being encoded in the frequency domain and one being encoded in the time domain.

10

In embodiments, the size of the data segments can be adapted to the size of the overlapping regions. Embodiments may have an efficient implementation of a combined MDCT/IMDCT processing, i.e. a block TDA comprising the operations of analysis windowing, folding and unfolding, and synthesis windowing. Moreover, in embodiments some of these steps may be partially or fully combined in an actual implementation.

15

20

Another embodiment of an apparatus 100 as shown in Fig. 1f illustrates that an apparatus 100 may further comprise a bypass 140 for the processor 120 and the overlay/add-combiner 130 being adapted for bypassing the processor 120 and the overlay/add-combiner 130 when non-overlapping consecutive time-domain data blocks occur in data segments, which are encoded in the time domain. If multiple data segments are encoded in the time domain, i.e. no conversion to the frequency domain may be necessary for decoding consecutive data segments, they may be transmitted without any overlapping. For these cases the embodiments as shown in Fig. 1f may bypass the processor 120 and the overlap/add-combiner 130. In embodiments the overlapping of blocks can be determined according to the AAC-specifications.

25

30

35

Fig. 2a shows an embodiment of an apparatus 200 for generating an encoded data stream based on a time-domain data stream, the time-domain data stream having samples of a signal. The time-domain data stream could, for example, correspond to an audio signal, comprising speech sections and music sections or both at the same time. The apparatus 200 comprises a segment processor 210 for providing data segments from the data stream, two consecutive data segments having a first or a second overlapping region, the second overlapping region being smaller than the first overlapping region. The apparatus 200 further comprises a time-domain encoder 220 for encoding a data segment in the time domain and a frequency-domain encoder 230 for applying weights to samples of the time-domain data stream according to a first or a second windowing function to obtain a windowed data segment, the first and second windowing functions being adapted to the first and second overlapping regions and for encoding the windowed data segment in the frequency domain.

Furthermore, the apparatus 200 comprises a time-domain data analyzer 240 for determining a transmission indication associated with a data segment and a controller 250 for controlling the apparatus such that for data segments having a first transition indication, output data of the time-domain encoder 220 is included in the encoded data stream and for data segments having a second transition indication, output data of the frequency-domain encoder 230 is included in the encoded data stream.

In embodiments the time-domain data analyzer 240 may be adapted for determining the transition indication from the time-domain data stream or from data segments provided by the segment processor 210. These embodiments are indicated in Fig. 2b. In Fig. 2b it is illustrated that the time-domain data analyzer 240 may be coupled to the input of the segment processor 210 in order to determine the transition indication from the time-domain data stream. In another

embodiment the time-domain data analyzer 240 may be coupled to the output of the segment processor 210 in order to determine the transition indication from the data segments. In embodiments the time-domain data analyzer 240 can be
5 coupled directly to the segment processor 210 in order to determine the transition indication from data provided directly by the segment processor. These embodiments are indicated by the dotted lines in Fig. 2b.

10 In embodiments the time-domain data analyzer 240 can be adapted for determining a transition measure, the transition measure being based on a level of transience in the time-domain data stream or the data segments wherein the transition indicator may indicate whether the level of
15 transience exceeds a predetermined threshold.

Fig. 2c shows another embodiment of the apparatus 200. In the embodiments shown in Fig. 2c the segment processor 210 can be adapted for providing data segments with the first
20 and the second overlapping regions, the time-domain encoder 220 can be adapted for encoding all data segments, the frequency-domain encoder 230 may be adapted for encoding all windowed data segments and the controller 250 can be adapted for controlling the time-domain encoder 220 and the
25 frequency-domain encoder 220 and the frequency-domain encoder 230 such that for data segments having a first transition indication, output data of the time-domain encoder 220 is included in the encoded data stream and for data segments having a
30 second transition indication, output data of the frequency-domain encoder 230 is included in the encoded data stream. In other embodiments both output data of the time-domain encoder 220 and the frequency-domain encoder 230 may be included in the encoded data stream. The transition
35 indicator may be indicating whether a data segment is rather associated or correlated with a speech signal or with a music signal. In embodiments the frequency-domain encoder 230 may be used for more music-like data segments

and the time-domain encoder 220 may be used for more speech-like data segments. In embodiments parallel encoding may be utilized, e.g. for a speech-like audio signal having background music.

5

In the embodiment depicted in Fig. 2c, multiple possibilities are conceivable for the controller 250 to control the multiple components within the apparatus 200. The different possibilities are indicated by dotted lines in Fig. 2c. For example, the controller 250 could be coupled to the time-domain encoder 220 and the frequency-domain encoder 230 in order to choose which encoder should produce an encoded output based on the transition indication. In another embodiment the controller 250 may control a switch at the outputs of the time-domain encoder 220 and the frequency-domain encoder 230.

In such an embodiment both the time-domain encoder 220 and the frequency-domain encoder 230 may encode all data segments and the controller 250 may be adapted for choosing via said switch which is coupled to the outputs of the encoders, which encoded data segment should be included in the encoded data stream, based on coding efficiency, respectively the transition indication. In other embodiments the controller 250 can be adapted for controlling the segment processor 210 for providing the data segments either to the time-domain encoder 220 or the frequency-domain encoder 230. The controller 250 may also control the segment processor 210 in order to set overlapping regions for a data segment. In other embodiments the controller 250 may be adapted for controlling a switch between the segment processor 210 and the time-domain encoder 220, respectively the frequency-domain encoder 230. The controller 250 could then influence the switch so to direct data segments to either one of the encoders, respectively to both. The controller 250 can be further adapted to set the windowing functions for the

frequency-domain encoder 230 along with the overlapping regions and coding strategies.

- Moreover, in embodiments the frequency-domain encoder 230
- 5 can be adapted for applying weights of window functions according to AAC specifications. The frequency-domain encoder 230 can be adapted for converting a windowed data segment to the frequency domain to obtain a frequency-domain data segment. Moreover, the frequency domain encoder
- 10 230 can be adapted for quantizing the frequency-domain data segments and, furthermore, the frequency-domain encoder 230 may be adapted for evaluating the frequency-domain data segments according to a perceptual model.
- 15 The frequency-domain encoder 230 can be adapted for utilizing a cosine modulated filterbank, an extended lapped transform, a low-delay filterbank or a polyphase filterbank to obtain the frequency-domain data segments.
- 20 The frequency-domain encoder 230 may be adapted for utilizing an MDCT to obtain the frequency data segments. The time-domain encoder 220 can be adapted for using a prediction model for encoding the data segments.
- 25 In embodiments where an MDCT in the frequency-domain encoder 230 operates in a so-called long block mode, i.e. the regular mode of operation that is used for coding non-transient input signals, compare AAC-specifications, the overhead introduced by the switching process may be high.
- 30 This can be true for the cases where only one frame, i.e. a length/framing rate of N samples, should be coded using the time-domain encoder 220 instead of the frequency-domain encoder 230.
- 35 Then all the input values for the MDCT may have to be encoded with the time-domain encoder 220, i.e. 2N samples are available at the output of the time-domain decoder 110. Thus, an overhead of N additional samples could be

introduced. Figs. 3a to 3d illustrate some conceivable overlapping regions of segments, respectively applicable windowing functions. $2N$ samples may have to be coded with the time-domain encoder 220 in order to replace one block of frequency-domain encoded data. Fig. 3a illustrates an example, where frequency-domain encoded data blocks use a solid line, and time-domain encoded data uses a dotted line. Underneath the windowing functions data segments are depicted which can be encoded in the frequency domain (solid boxes) or in the time domain (dotted boxes). This representation will be referred to in Figs. 3b to 3d as well.

Fig. 3a illustrates the case where data is encoded in the frequency domain, interrupted by one data segment which is encoded in the time domain, and the data segment after it is encoded in the frequency domain again. In order to provide the time-domain data which is necessary to cancel the time-domain aliasing evoked by the frequency-domain encoder 230, when switching from the frequency domain to the time domain, half of a segment size of overlapping is required, the same holds from switching back from the time domain to the frequency domain. Assuming that the time-domain encoded data segment in Fig. 3a has a size of $2N$, then at its start and at the end it overlaps with the frequency-domain encoded data by $N/2$ samples.

In case more than one subsequent frames can be encoded using the time-domain encoder 220, the overhead for the time-domain encoded section stays at N samples. As it is illustrated in Fig. 3b where two consecutive frames are encoded in the time domain and the overlapping regions at the beginning and the end of the time-domain encoded sections have the same overlap as it was explained with respect to Fig. 3a. Fig. 3b shows the overlap structure in case of two frames encoded with time-domain encoder 220. $3N$ samples have to be coded with the time-domain encoder 220 in this case.

This overhead can be reduced in embodiments by utilizing window switching, for example, according to the structure which is used in AAC. Fig. 3c illustrates a typical
5 sequence of Long, Start, 8Short and Stop windows, as they are used in AAC. From Fig. 3c it can be seen that the window sizes, the data segment sizes and, consequently, the size of the overlapping regions change with the different windows. The sequence depicted in Fig. 3c is an example for
10 the sequence mentioned above.

Embodiments should not be limited to windows of the size of AAC windows, however, embodiments take advantage of windows with different overlapping regions and also of windows of
15 different durations. In embodiments transitions to and from short windows may utilize a reduced overlap as, for example, disclosed in Bernd Edler, "Codierung von Audiosignalen mit überlappender Transformation und adaptiven Fensterfunktionen", Frequenz, Vol. 43, No. 9, p.
20 252-256, September 1989 and Generic Coding of Moving Pictures and Associated Audio: Advanced Audio Coding, International Standard 13818-7, ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group, 1997 may be used in
embodiments to reduce the overhead for the transitions to
25 and from the time-domain encoded regions, as it is illustrated in Fig. 3d. Fig. 3d illustrates four data segments, of which the first two and the last one are encoded in the frequency domain and the third one is encoded in the time domain. When switching from the
30 frequency domain to the time domain different windows with the reduced overlapping size are used, therewith reducing the overhead.

In embodiments the transition may be based on Start and
35 Stop windows identical to the ones used in AAC. The corresponding windows for the transitions to and from the time-domain encoded regions are windows with only small regions of overlap. As a consequence, the overhead, i.e.

the number of additional values to be transmitted due to the switching process decreases substantially. Generally, the overhead may be $N_{ovl}/2$ for each transition with the window overlap of N_{ovl} samples. Thus, a transition with the regular fully-overlapped window like an AAC with $N_{ovl} = 1024$ incurs an overhead of $1024/2 = 512$ samples for the left, i.e. the fade-in window, and $1024/2 = 512$ samples for the right, i.e. the fade-out window, transition resulting in a total overhead of 1024 (= N) samples. Choosing a reduced overlap window like the AAC Short block windows with $N_{ovl}=128$ only results in an overall overhead of 128 samples.

Embodiments may utilize a filterbank in the frequency-domain encoder 230 as, for example, the widely used MDCT filterbank, however, other embodiments may also be used with frequency-domain codecs based on other cosine-modulated filterbanks. This may comprise the derivatives of the MDCT, such as extended lapped transforms or low-delay filterbanks as well as polyphase filterbanks, such as, for example, the one used in MPEG-1-Layer-1/2/3 audio codecs. In embodiments efficient implementation of a forward/back-filterbank operation may take into account a specific type of window and folding/unfolding used in the filterbank. For every type of modulated filterbank the analysis stage may be implemented efficiently by a preprocessing step and a block transform, i.e. DCT-like or DFT, for the modulation. In embodiments the corresponding synthesis stage can be implemented using the corresponding inverse transform and a post processing step. Embodiments may only use the pre- and post processing steps for the time-domain encoded signal portions.

Embodiments of the present invention provide the advantage that a better code efficiency can be achieved, since switching between a time-domain encoder 220 and the frequency-domain encoder 230 can be done introducing very low overhead. In signal sections of subsequent time-domain encoding only, overlap may be omitted completely in

embodiments. Embodiments of the apparatus 100 enable the according decoding of the encoded data stream.

5 Embodiments therewith provide the advantage that a lower coding rate can be achieved for the same quality of, for example, an audio signal, respectively a higher quality can be achieved with the same coding rate, as the respective encoders can be adapted to the transience in the audio signal.

10

Depending on certain implementation requirements of the inventive methods, the inventive methods can be implemented in hardware or in software. The implementation can be performed using a digital storage medium, in particular a
15 disc, DVD or CD having electronically stored control signals stored thereon, which cooperate with the programmable computer system such that the inventive methods are performed. Generally, the present invention is, therefore, a computer program product having a program code
20 stored on a machine-readable carrier, the program code being operative for performing the inventive methods when the computer program product runs on a computer. In other words, the inventive methods are, therefore, a computer program having a program code for performing at least one
25 of the inventive methods when the computer program runs on a computer.

Reference List

- 100 apparatus for decoding
- 110 time-domain decoder
- 5 120 processor
- 122 frequency-domain decoder
- 122a re-quantization
- 124 time-domain to frequency-domain converter
- 124a modified discrete cosine transform
- 10 126 frequency-domain combiner
- 126a adder
- 128 frequency-domain to time-domain converter
- 128a inverse modified discrete cosine transform
- 129 calculator
- 15 129a time-domain aliasing stage
- 130 overlap/add-combiner
- 200 apparatus for encoding
- 210 segment processor
- 220 time-domain encoder
- 20 230 frequency-domain encoder
- 240 time-domain data analyzer
- 250 controller
- 400 modified discrete cosine transform input
- 410 windows
- 25 420 inverse modified discrete cosine transform output
first window
- 425 inverse modified discrete cosine transform output
second window
- 430 final output

Claims

1. An apparatus for decoding data segments representing a time-domain data stream, a data segment being encoded in the time domain or in the frequency domain, a data segment being encoded in the frequency domain having successive blocks of data representing successive and overlapping blocks of time-domain data samples, wherein two consecutive time-domain data blocks of which one being encoded in the frequency domain and one being encoded in the time domain having a time domain overlapping region, the apparatus comprising:

a time-domain decoder for decoding a data segment being encoded in the time domain to obtain output data of the time-domain decoder;

a processor comprising a frequency-domain decoder for decoding data segments being encoded in the frequency domain to obtain frequency-domain data segments and for processing the frequency-domain data segments and the output data of the time-domain decoder to obtain overlapping time-domain data blocks for the time domain overlapping region; and

an overlap/add-combiner for combining the overlapping time-domain data blocks in the time domain overlapping region to obtain a decoded data segment of the time-domain data stream for the time domain overlapping region.
2. The apparatus of claim 1, wherein the processor is adapted for processing a data segment being encoded in the time domain and in the frequency domain in parallel.

3. The apparatus of claim 1, wherein the processor comprises a time-domain to frequency-domain converter for converting the output data of the time-domain decoder to obtain converted frequency-domain data segments.
4. The apparatus of claim 3, wherein the processor comprises a frequency-domain combiner for combining the frequency-domain data segments and the converted frequency-domain data segments to obtain a frequency-domain data stream.
5. The apparatus of claim 4, wherein the processor comprises a frequency-domain to time-domain converter for converting the frequency-domain data stream to overlapping time-domain data blocks.
6. The apparatus of claim 1, wherein the frequency domain decoder further comprises a re-quantization stage.
7. The apparatus of claim 3, wherein the time-domain to frequency-domain converter comprises a cosine modulated filterbank, an extended lapped transform, a low-delay filterbank, a polyphase filterbank or a modified discrete cosine transform.
8. The apparatus of claim 4, wherein the frequency-domain combiner comprises an adder.
9. The apparatus of claim 5, wherein the frequency-domain to time-domain converter comprises a cosine modulated filterbank or an inverse modified discrete cosine transform.
10. The apparatus of claim 1, wherein the time-domain decoder is adapted for using a prediction filter to decode a data segment encoded in the time domain.

11. The apparatus of claim 1, wherein the processor comprises a calculator for calculating overlapping time-domain data blocks based on the output data of the time-domain decoder.
12. The apparatus of claim 11, wherein the calculator is adapted for reproducing an overlapping property of the frequency-domain to time-domain converter based on the output data of the time-domain decoder.
13. The apparatus of claim 12, wherein the calculator is adapted for reproducing a time-domain aliasing characteristic of the frequency-domain to time-domain converter based on the output data of the time-domain decoder.
14. The apparatus of claim 5, wherein the frequency-domain to time-domain converter is adapted for converting the frequency-domain data segments provided by the frequency-domain decoder to overlapping time-domain data blocks.
15. The apparatus of claim 14, wherein the overlap/add-combiner is adapted for combining the overlapping time-domain data blocks provided by the frequency-domain to time-domain converter and the calculator to obtain decoded data segments of the time-domain data stream.
16. The apparatus of claim 7, wherein the calculator comprises a time-domain aliasing stage for time-aliasing output data of the time-domain decoder to obtain the overlapping time-domain data blocks.
17. The apparatus of claim 11, wherein the calculator is adapted for

segmenting the output of the time-domain decoder in calculator segments comprising $2N$ sequential samples,

applying weights to the $2N$ samples according to an analysis window function,

subtracting the first $N/2$ samples in reversed order from the second $N/2$ samples,

adding the last $N/2$ samples in reversed order to third $N/2$ samples,

inverting the second and third $N/2$ samples

replacing the first $N/2$ samples with the time-reversed and inverted version of the second $N/2$ samples,

replacing the fourth $N/2$ samples with the time-reversed version of the third $N/2$ samples, and

applying weights to the $2N$ samples according to a synthesis windowing function.

18. The apparatus of claim 5, wherein the overlap/add-combiner is adapted for applying weights according to a synthesis windowing function to overlapping time-domain data blocks provided by the frequency-domain to time-domain converter.
19. The apparatus of claim 18, wherein the overlap/add-combiner is adapted for applying weights according to a synthesis windowing function being adapted to a size of an overlapping region of consecutive overlapping time-domain data blocks.
20. The apparatus of claim 19, wherein the calculator is adapted for applying weights to the $2N$ samples according to an analysis windowing function being

adapted to a size of an overlapping region of consecutive overlapping time-domain data blocks and wherein the calculator is adapted for applying weights to the $2N$ samples according to a synthesis windowing function being adapted to the size of the overlapping region.

21. The apparatus of claim 1, wherein a size of an overlapping region of two consecutive time-domain data blocks which are encoded in the frequency domain is larger than a size of an overlapping region of two consecutive time-domain data blocks of which one being encoded in the frequency domain and one being encoded in the time domain.
22. The apparatus of claim 1, wherein the overlapping of data blocks is being determined according to the AAC-specifications.
23. The apparatus of claim 1, further comprising a bypass for the processor and the overlap/add-combiner, the bypass being adapted for bypassing the processor and the overlap/add-combiner when non-overlapping consecutive time-domain data blocks incur in data segments which are encoded in the time domain.
24. Method for decoding data segments representing a time-domain data stream, a data segment being encoded in the time domain or in the frequency domain, a data segment being encoded in the frequency domain having successive blocks of data representing successive and overlapping blocks of time-domain data samples, wherein two consecutive time-domain data blocks of which one being encoded in the frequency domain and one being encoded in the time domain having a time domain overlapping region, comprising the steps of:

decoding a data segment being encoded in the time domain to obtain output data of the time-domain decoder;

decoding data segments being encoded in the frequency domain to obtain frequency-domain data segments;

processing the frequency-domain data segments and the output data of the time-domain decoder to obtain overlapping time-domain data blocks for the time domain overlapping region; and

combining the overlapping time-domain data blocks in the time domain overlapping region to obtain a decoded data segment of the time-domain data stream for the time domain overlapping region.

25. Computer program having a program code for performing the method of claim 24, when the program code runs on a computer.

26. An apparatus for generating an encoded data stream based on a time-domain data stream, the time-domain data stream having samples of a signal, the apparatus comprising

a segment processor for providing data segments from the data stream, two consecutive data segments having a first or a second overlapping region, the second overlapping region being smaller than the first overlapping region;

a time-domain encoder for encoding a windowed data segment in the time domain;

a frequency-domain encoder for applying weights to samples of the time-domain data stream according to a first or second windowing function to obtain a windowed data segment, the first and second windowing functions being adapted to the first and second overlapping regions, the frequency-domain encoder

being adapted for encoding a windowed data segment in the frequency domain;

a time-domain data analyzer for determining a transition indication associated with a data segment; and

a controller for controlling the apparatus such that for data segments having a first transition indication output data of the time-domain encoder is included in the encoded data stream and for data segments having a second transition indication, output data of the frequency-domain encoder is included in the encoded data stream.

27. The apparatus of claim 26, wherein the time-domain data analyzer is adapted for determining the transition indication from the time-domain data stream, the data segments or from data directly provided by the segment processor.
28. The apparatus of claim 26, wherein the time-domain data analyzer is adapted for determining a transition measure, the transition measure being based on the level of transience in the time-domain data stream or the data segment and wherein the transition indicator indicates whether a level of transience exceeds a predetermined threshold.
29. The apparatus of claim 26, wherein the segment processor is adapted for providing data segments with the first and the second overlapping regions,

the time-domain encoder is adapted for encoding the data segments,

the frequency-domain encoder is adapted for encoding the windowed data segments, and

the controller is adapted for controlling the time-domain encoder and the frequency-domain encoder such that for data segments having a first transition indication output data of the time-domain encoder is included in the encoded data stream and for windowed data segments having a second transition indication output data of the frequency-domain encoder is included in the encoded data stream.

30. The apparatus of claim 26, wherein the controller is adapted for controlling the segment processor for providing the data segments either to the time-domain encoder or the frequency-domain encoder.
31. The apparatus of claim 26, wherein the frequency-domain encoder is adapted for applying weights of windowing functions according to the AAC-specifications.
32. The apparatus of claim 26, wherein the frequency-domain encoder is adapted for converting a windowed data segment to the frequency domain to obtain a frequency-domain data segment.
33. The apparatus of claim 32, wherein the frequency-domain encoder is adapted for quantizing the frequency-domain data segment.
34. The apparatus of claim 33, wherein the frequency-domain encoder is adapted for evaluating the frequency-domain data segment according to a perceptual model.
35. The apparatus of claim 34, wherein the frequency-domain encoder is adapted for utilizing a cosine-modulated filterbank, an extended lapped transform, a

low-delay filterbank or a polyphase filterbank to obtain the frequency-domain data segments.

36. The apparatus of claim 32, wherein the frequency-domain encoder is adapted for utilizing a modified discrete cosine transform to obtain the frequency-domain data segments.
37. The apparatus of claim 26, wherein the time-domain encoder is adapted for using a prediction filter for encoding the data segments.
38. Method for generating an encoded data stream based on a time-domain data stream, the time-domain data stream having samples of a signal, comprising the steps of

providing data segments from the data stream, two consecutive data segments having a first or a second overlapping region, the second overlapping region being smaller than the first overlapping region;

determining a transition indication associated with the data segments;

encoding a data segment in the time domain, and/or

applying weights to samples of the time-domain data stream according to a first or a second windowing function to obtain a windowed data segment, the first and second windowing functions being adapted to the first and second overlapping regions and encoding the windowed data segment in the frequency domain and;

controlling such that for data segments having a first transition indication output data being encoded in the time-domain is included in the encoded data stream and for data segments having a second transition

indication output data being encoded in the frequency domain is included in the encoded data stream.

39. Computer program having a program code for performing the method of claim 38, when the program code runs on a computer.