Audio Transmitter Processor, Audio Receiver Processor and Related Methods and Computer Programs

Specification

The present invention is related to audio processing and, particularly, to audio processing applications that are useful in the context of error-prone transmission channels such as wireless channels.

US Patent 5,852,469 discloses a moving picture coding and/or decoding system and a variable length coding and/or decoding system. The encoder has a divider for dividing a code stream supplied from the encoder into a plurality of code strings and a reorderer for arranging at least one of the plurality of code strings in the forward direction from the head to the end and at least one of the other code strings in the backward direction from the end to the head. A variable-length coding system includes a codeword table for storing a plurality of codewords so that the codewords correspond to source symbols. And an encoder selects a codeword corresponding to the source symbol input from the codeword table and for outputting the selected codeword as coded data. The plurality of codewords can be decoded in either of the forward and backward directions. The plurality of codewords are configured so that the pause between codes can be detected by a predetermined weight of the codeword, in the case of a binary code, the number of "1" or "0" in the codeword.

EP 1155498 B1 discloses a concept for producing or reading a data stream that comprises a multitude of raster points as reference points, wherein at least a part of each codeword of a first set is written in a first direction of writing starting at the raster point of a segment, and at least a part of a codeword of a second set of codewords is written into the data stream in a second direction of writing which is opposite to the first direction of writing, starting from a second raster point of a segment. In case that a codeword of the second set does not or not completely fit into a segment, at least a part of this codeword or a part of the remainder of this codeword which does not fit into the assigned segment is written into a different, not fully occupied segment, in accordance with a predetermined rule.

This procedure makes sure that an error propagation is limited to the space between two raster points.

MPEG-4 Part 3 Audio [1] defines BSAC, which uses bit sliced arithmetic coding, where psychoacoustic relevancy decreases linearly over the audio frame.

MPEG-4 Part 3 Audio [1] defines error sensitivity categories for the bitstream payload of AAC (Table 4.94 in [1]):

category	payload	mandatory	leads / may lead to one instance per	Description
0	main	yes	CPE / stereo layer	commonly used side information
1	main	yes	ICS	channel dependent side information
2	main	no	ICS	error resilient scale factor data
3	main	no	ICS	TNS data
4	main	yes	ICS	spectral data
5	extended	no	EPL	extension type / data_element_version
6	extended	no	EPL	DRC data
7	extended	no	EPL	bit stuffing
8	extended	no	EPL	ANC data
9	extended	no	EPL	SBR data

Related data is subsequently stored in instances of corresponding error sensitivity classes to form an ER AAC payload, which may subsequently be protected individually using forward error correction or detection means. A fix assignment of data elements into categories is specified. Due to entropy coding, this leads to classes of variable lengths. Those lengths need to be transmitted to allow the decoding of the ER AAC payload, which causes additional overhead.

DRM [2] defines super frames for the bitstream payload of AAC to allow unequal error protection (aac_super_frame). A super frame consists of a predefined number (either 5 or 10) AAC frames. It is assumed, that the psychoacoustically more important bits of an AAC frame are available at the beginning of the bitstream payload. Therefore, the first N bits (e.g. 200 bits) are cut from each frame and are consecutively stored at the beginning of the super frame. Those bits are subsequently protected by a CRC. The remaining bits of those frames are stored afterwards without protection. Since always a fix amount of data is treated as sensitive, no length information needs to be transmitted in order to decode the protected payload (of course, lengths information for the individual frames of a super frame is needed, but this is out of scope for the current consideration).

The frame generated by BSAC as described in MPEG-4 Part 3 Audio comes already sorted by psychoacoustic relevancy; it starts with the most important bits and ends with the least important bits. This comes by the cost of higher computational complexity for arithmetical en-/decoding of all bits of the spectral lines.

Due to the nature of the approach for AAC as described in MPEG-4 Part 3, the instances of the various error sensitivity categories are of variable lengths. This is no issue for convolutional codes, but is inappropriate for block codes, which require a fix amount of data to be protected.

The DRM approach just works, if the bitstream payload is already arranged based on the psychoacoustic importance of the individual bits.

It is an object of the present invention to provide an improved and nevertheless efficient concept for generating an error protected frame or for processing a received error protected frame.

This object is achieved by an audio transmitter processor of claim 1, an audio receiver processor of claim 23, a method of audio transmission processing of claim 44, a method of audio receiving processing of the claim 45, or a computer program of claim 46.

An audio transmitter processor for generating an error protected frame uses encoded audio data corresponding to an audio frame, where this encoded audio data comprises a first amount of information units such as bits or bytes and a second amount of information units. A frame builder builds a frame having a codeword raster defining reference positions for a predefined total number of codewords, where the frame builder is configured to write the information units of the first amount of information units starting at reference positions of a first predefined subset of the codewords and to write the information units of the second amount of information units starting at reference positions of a second predefined subset of the codewords, where the frame builder determines a border between the first amount of information units and the second amount of information units so that a starting information unit of the second amount of information units coincides with a codeword border. The audio transmitter processor has an error protection coder for processing the predefined total number of codewords individually to obtain a plurality of processed codewords representing the error protected frame and/or for

processing one or more of the codewords of the first predefined subset to obtain a first processing result and/or for processing one or more of the codewords of the second predefined subset to obtain a second processing result and for adding the first processing result or the second processing result to the predefined number of codewords to obtain the plurality of processed codewords.

On the receiver side, an audio receiver processor for processing a received error protected frame comprises a receiver interface for receiving the error protected frame. The audio receiver processor comprises an error protection processor for processing the error protected frame to obtain an encoded audio frame. Particularly, the error protection processor is configured to check whether a first predefined subset of the codewords of the encoded audio frame comprises an error. The audio receiver processor comprises an error concealer or error concealment indicator configured to perform a (full) frame loss concealment operation in case of a detected error in the first predefined subset of the codewords or to generate and forward an error concealment indication indicating the frame loss concealment operation to be done at a remote place.

Due to the separate processing of the first predefined subset of the codewords on the one hand and the second predefined subset of the codewords on the other hand and by using the information on the first predefined subset of the codewords on the receiver side, a very efficient processing with respect to the generation of an error protected frame and a processing with respect to the error checking is obtained, since the predefined subset of the codewords of the first set is predefined and, therefore, known to the decoder without any specific additional signalization such as signalization bit per frame or so. This is not required; instead, because the encoder uses a predefined subset of first codewords for writing the first amount of information units and since the receiver or audio receiver processor relies on this predefinition, an efficient error protection on the one hand and efficient error checking on the other hand is made available.

Preferably, the error protection processing on the reception side allows a separate calculation of a processing result such as a Hash value on two or more of the first subset of the codewords but not any codeword from the second set and, at the same time, the calculation of a Hash value only from the codewords of the second predefined subset of the codewords without any codewords from the first set allows an efficient error checking processing on the decoder side, since only a certain amount rather than all codewords must be used for Hash verification. Very early in the receiver processing, it can be

determined whether serious errors have occurred in the frame that, in the end, result in a requirement for a full frame loss concealment operation, or whether only relatively less important audio data have been affected by transmission errors so that only a much higher quality partial frame loss concealment operation or no concealment operation at all is necessary for addressing this type of error.

Due to the fact that the present invention forms a bridge between audio encoding on the one hand and error protection processing on the other hand via the specific frame building operation, very efficient and very high quality and smart error processing procedure can be applied on the decoder side due to the separate error protection processing for the first predefined subset of the codewords having the first amount of data and the second predefined subset of the codewords having the second amount of data. Preferably, the first amount of data are psychoacoustically more important data or are side information and optional TNS data and most and least significant bits of lower spectral values while the second amount of data typically comprises most and least significant bits of higher frequencies that are not so decisive for the audio perception from a psychoacoustic point of view. Further information units that are typically in the second amount of information units are residual data that are generated provided that the bit consumption by the arithmetic encoder has not fully consumed the available bit budget.

Particularly, the writing of the first amount of information units and the second amount of information units into first and second predefined subsets, where a border between the first amount of information units and the second amount of information units is placed at a codeword border makes sure that a clear separation is found between codewords that are more important, i.e., the first predefined subset of the codewords compared to codewords that are less important such as the second predefined subset of the codewords. In a scenario where the coding operation applied by the audio coder is a signal-dependent coding operation that, in the end, results in a variable length result of audio data for a frame that is adapted to a fixed frame raster by controlling the coding operation and by calculating additional residual bits, for example, the border between the first amount of information units and the second amount of information units dynamically changes from frame to frame. Nevertheless, the psychoacoustically more important data such as low frequency data are included in the first predefined subset and, therefore, on the transmitter side, only a check of the first predefined subset of the codewords results in a situation, where a full frame loss concealment is to be performed while, as soon as it has been determined on the receiver side that the first predefined subset of the codewords

has been received without any errors, only then a further processing such as a check of the second predefined subset of the codewords is to be done. Therefore, as soon as it is determined that the first predefined subset of the codewords has an error, a full frame loss concealment operation such as a repetition of an earlier frame or a modified repetition of a preceding frame or anything like that is performed without spending any resources for further processing the received erroneous frame.

The receiver processor comprises a frame reader for reading the encoded audio frame in accordance with a predefined frame reading procedure identifying the first predefined subset of the codewords and the second predefined subset of the codewords. Any audio data processing order that has been applied by the encoder side frame builder can be undone/rearranged or is, for a direct reading procedure, known to the decoder so that the decoder can parse the received frame at least with respect to the first predefined subset, when an error-free condition has been detected for this first predefined subset and even for the second predefined subset, in case an error-free condition of the second predefined subset has been determined as well.

The frame reader typically only has to be activated subsequent to the determination of an error-free situation of the first predefined subset of the codewords. The error protection processor only has to know the location of the first predefined subset of the codewords in the data frame output by the error protection processor but does not have to know, for the purpose of error checking, in which directions any data has been written into the corresponding positions represented by the codewords.

Preferably, psychoacoustically less important data are located at specific positions in the frame which can be at the left border of the frame or at the right border of the frame or at a predefined number of codewords/reference positions within the frame. It is desirable to separate psychoacoustically more important data from the psychoacoustically less important data or it is required to rearrange psychoacoustically more important data and psychoacoustically less important data within an audio frame.

A rearrangement is, for example, necessary to align the data to a given error protection and detection scheme, when the frame of encoded audio data is generated by a predefined and standardized audio decoder that is not yet customized to cooperate with a certain error protection processor. This rearrangement allows individual frame loss

concealment procedures depending on the availability of the psychoacoustically more important data and the psychoacoustically less important data.

Preferred embodiments of the present invention are subsequently discussed with respect to the accompanying drawings in which:

Fig. 1 is an illustration of an example of an originally LC3 bitstream payload;

Fig. 2 illustrates a distribution of bits of the example LC3 bitstream payload given in Fig. 1 based on their psychoacoustic relevancy;

Fig. 3 illustrates an example of an LC3 bitstream payload rearrangement;

Fig. 4 illustrates another example of an LC3 bitstream payload rearrangement with an arithmetic encoder/decoder operating on byte granularity;

Fig. 5 illustrates a preferred implementation of an audio transmitter processor;

Fig. 6 illustrates a procedure for implementing the frame building;

Fig. 7 illustrates a preferred procedure performed by the frame builder of Fig. 5;

Fig. 8 illustrates the preferred procedure of the frame builder;

Figs. 9a-9c illustrate schematic representations of locations of the first predefined subset of the codewords and the second predefined subset of the codewords within a frame built by the frame builder of Fig. 5;

Fig. 10 illustrates a preferred implementation of a direct writing of the frame by the frame builder;

Fig. 11 illustrates a preferred implementation of the rearrangement procedure of the frame builder of Fig. 5;

Fig. 12 illustrates a preferred implementation of the error protection coder of Fig. 5;

Fig. 13 illustrates a preferred implementation of the audio receiver processor in accordance with the present invention;

Fig. 14 illustrates a preferred procedure of the error protection processor and the error concealer;

Fig. 15 illustrates a further preferred implementation of the error protection processor and the error concealer;

Fig. 16 illustrates a schematic representation of the concealment spectral range for a partial frame loss concealment;

Fig. 17 illustrates a further implementation of a partial frame loss concealment;

Fig. 18 illustrates a preferred implementation of the frame reader of Fig. 13;

Fig. 19 illustrates a preferred implementation of the frame reader for performing a rearrangement of received data into a frame format required by a specific standard, such as standardized audio decoder;

Fig. 20 illustrates a preferred procedure done by the frame reader for a direct reading of the audio data of the error protected frame;

Fig. 21 illustrates a preferred implementation of an audio encoder of Fig. 5; and

Fig. 22 illustrates a preferred implementation of an audio decoder of Fig. 13.

Subsequently, preferred implementations of the present invention in certain contexts are discussed.

The bits are written chronologically - but not spatially - during the encoding process based on their psychoacoustic relevancy. The most important data are written first, the least important data are written last. However, the position of the psychoacoustically less important bits within a 'normal' audio frame may vary from frame to frame depending on the underlying coded data. This might be for example due to writing the data into the frame from both sides, whereas from one side arithmetically coded data is written and

from the other side data coded by other means is written simultaneously. An example for such an approach is the LC3 codec.

A system with two classes of bits is envisioned. The subdivision of the bitstream payload into two classes is done based on their relevancy relative to the output:

- Bits, which are psychoacoustically less important – their distortion allows partial frame loss concealment, are put into one class;
- Bits, which are psychoacoustically more important – their distortion requires full frame loss concealment, are put into another class.

Reason for doing so is that – depending on the availability of the two classes – different concealment strategies are envisioned. Those two different concealment strategies are referred to subsequently as full frame loss concealment and partial frame loss concealment:

- Full frame loss concealment takes place, if the class covering the psychoacoustically more important bits is lost. The availability of the class covering the psychoacoustically less important bits does not matter – its data is not evaluated. In that case, no data of the current frame is available, so the frame is synthesized completely based on the last received frame.
- Partial frame loss concealment may take place, if the class covering the psychoacoustically more important bits is available, but the class covering the psychoacoustically less important data is lost. In that case, the psychoacoustically more important data is available and can be used to reconstruct the frame – just the psychoacoustically less important data needs to be synthesized based on the last fully received frame. Partial frame loss concealment is meant to provide better quality (i.e. less artifacts) than full frame loss concealment under many circumstances.

The sizes of the two classes are predefined, e.g. by the channel coder.

The forward error detection and correction scheme preferably utilizes Reed-Solomon codes and works on a codeword granularity, whereas each codeword consists of multiple nibbles (4 bits, also called semi-octets). In the present case (LC3), one codeword consists

of 13 to 15 nibbles. In a preferred embodiment, such forward error detection and correction scheme offers various degrees of error detection and correction, depending on the overhead being spent, e. g.

- 4 nibbles overhead per codeword allow 2 nibbles to be corrected (error protection mode 3);

- 6 nibbles overhead per codeword allow 3 nibbles to be corrected (error protection mode 4).

For a given gross bitrate, the net bitrate depends on the chosen error protection mode – the higher the error detection and correction capability, the smaller the available net bitrate.

Individual error detection is required for both classes. Considering the given channel coding configuration, it is preferred to store all bits of one class into a certain number of codewords, and all bits of the other class into the remaining number of codewords.

The subdivision of the bits within the bitstream payload into classes is made such, that always a certain number of codewords comprises the bits of one class, whereas the remaining number of codewords comprises the bits of the other class.

As stated before, the position of the psychoacoustically less important bits may vary from frame to frame depending on the underlying coded data.

However, the goal is to have a certain amount of psychoacoustically less important bits separated for rearrangement (allowing individual error protection and detection in combination with fix codeword sizes and positions provided by the channel codec).

The rearrangement has to be done adaptively on a frame-by-frame basis. For such rearrangement, no additional side information (e.g. length information) shall be required to revert the rearrangement or to decode the rearranged bitstream payload.

A bitstream payload is usually written such, that writing and reading routines can be implemented in an efficient way. The psychoacoustic importance plays usually no role, which may lead to a bitstream payload, where psychoacoustically less important bits and

psychoacoustically more important bits are mixed. In order to allow an efficient forward error protection in the context of the given requirements mentioned before, such a bitstream payload is suboptimal and requires rearrangement.

Since the position of the psychoacoustically less important bits may vary from frame to frame depending on the underlying coded data, no direct mapping to the fixed codeword sizes and positions is possible. Therefore, – in a straightforward approach - the related bits are rearranged as follows:

- psychoacoustically less important bits are stored at one end of the presorted bitstream payload.

- psychoacoustically more important bits are stored at the other end of the presorted bitstream payload.

The number of the psychoacoustically less important bits is statically derived based on an error protection mode and the size of the channel-encoded frame.

However, the location of the psychoacoustically less important bits is dynamically derived. Adaptive rearrangement rules are predefined, such that no additional side information is needed to repeal the rearrangement at the decoder side. The adaptation rules make sure, that the psychoacoustically least important bits are always stored at the far end of the chosen side of the bitstream, and assure at the same time, that the decoder knows exactly, how to restore the original bitstream payload.

On one hand, such rearrangement can be done as a post-processing step, after the 'normal' bitstream payload has been completely written by the encoder – and as a pre-processing step after decoding the side information (which is never part of the rearrangement), before the remaining 'normal' payload is read by the decoder.

On the other hand, such rearrangement can also be done during the encoding process, writing the encoded bits directly at the appropriate position – and during the decoding process, reading the bits directly from the appropriate position.

It is noted, that any assignment of the psychoacoustically less and the psychoacoustically more important bits to dedicated codewords is possible. The assignment of the

psychoacoustically less important bits to the left-most codewords and the assignment of the psychoacoustically more important bits to the right-most codewords is just one preferred embodiment. Accordingly, the rearrangement could also be done differently, depending on the chosen assignment. The only prerequisite is that the assignment is predefined, such that the decoder can revert the process without additional information in the bitstream.

The following application scenarios are considerable:

1. The audio frame shall be written such, that the psychoacoustically less important bits are collected on one end of the bitstream:
    a. Start writing into the area provided for the more significant bits from both sides.
    b. Stop, when the two writing pointers meet each other.
    c. Continue writing from both sides into the area provided for the less significant bits.

2. The audio frame is written in the 'normal' way, but shall be rearranged such, that the psychoacoustically less important bits are collected on one end of the bitstream:
    a. Start reading and parsing the frame from both sides.
    b. Stop, when the amount of bits provided for the more significant bits is read: The psychoacoustically less important bits are the remaining bits between the two reading pointers.
    c. Exchange the data up to the left pointer with the data between the two pointers.

3. The audio frame is written such, that the psychoacoustically less important bits are collected on one end of the bitstream, but shall be rearranged in the 'normal' way:
    a. Start reading the frame from both sides. If the psychoacoustically less important bits should be stored at the left side of the bitstream, the starting point for reading the psychoacoustically more important bits from the left side can be derived from (means: is equal to) the number of bits provided for the psychoacoustically less important bits. If the psychoacoustically less important bits should be stored at the right side of the bitstream, the starting point for reading the psychoacoustically more important bits from

the right side can be derived from the number of bits provided for the psychoacoustically less important bits and the number of total bits.

b. Stop, when the two writing pointers meet each other.

c. Exchange the data up to the starting point for reading the psychoacoustically more important bits (see a.) with the data between this starting point and the meeting point of the two pointers.

4. The audio frame written such, that the psychoacoustically less important bits are collected on one end of the bitstream, shall be read:

a. Start reading the frame from both sides. If the psychoacoustically less important bits should be stored at the left side of the bitstream, the starting point for reading the psychoacoustically more important bits from the left side can be derived from the number of bits provided for the psychoacoustically less important bits. If the psychoacoustically less important bits should be stored at the right side of the bitstream, the starting point for reading the psychoacoustically more important bits from the right side can be derived from the number of bits provided for the psychoacoustically less important bits and the number of total bits.

b. Stop, when the two writing pointers meet each other.

c. Continue reading from the remaining part of the frame both sides. The starting point for reading the psychoacoustically less important bits from the right side is the same as the starting point for reading the psychoacoustically more important bits from the left side (see a.).

Fig. 5 illustrates an audio transmitter processor in accordance with a preferred embodiment of the present invention. The audio transmitter processor preferably comprises a frame builder 506 and an error protection coder 508.

The input into the audio transmitter processor is an amount of encoded audio data such as audio data derived from a frame of audio data input into an audio encoder 500 that, typically, is a variable length audio encoder. The bit amount required by the audio encoder depends on the signal to be encoded, but, in a preferred implementation, the output of audio data – typically in the form of a frame of encoded audio data – is a fixed length frame. Therefore, the audio encoder typically encodes with variable quality so that a frame of audio data to be encoded that is difficult to encode is, in the end, represented in the encoded audio data by a lower quality representation while a frame of audio data to be

encoded that is easy to encode, i.e., that can be encoded with a lower number of bits for a certain quality level is represented, in the end, at the output of the audio encoder 500 by a higher quality representation.

Typically, the output of the audio encoder for a certain, for example time domain portion of audio data comprises a first amount of information units and a second amount of information units. In case of a 50 percent overlap add situation, the time domain portion has twice the size of a frame, i.e., a number of audio samples newly input into the encoder or output by an overlap add stage of a decoder.

The frame builder 506 in Fig. 5 that may, for example, comprise a rearranger 502 and a subsequently connected frame writer 504 or that may, for example, consist of a frame writer only in case any intermediate representation is not an issue, is configured for building a frame having a codeword raster defining reference positions for a total number of predefined codewords for the frame. Such a frame is, for example, illustrated with respect to Figs. 9a, 9b, 9c, where different codeword arrangements are given and where, for example, reference positions for the predefined codewords are indicated as vertical lines and those reference positions are a start of a codeword or an end of a codeword or both. These reference positions or raster points do not require any specific information units but are given by a certain bit or byte position, to which some data is written. Hence, the reference positions or raster points do not incur any overhead and typically mark the begin or end of a codeword for the error protection processor.

In accordance with the present invention, the frame builder is configured to write the information units of the first amount of information units starting at reference positions of a first predefined subset of the codewords.

The information units of the second amount of information units are written starting at reference positions of a second predefined subset of the codewords. Particularly, the frame builder 506 is configured to determine a border between the first amount of information units and the second amount of information units so that a starting information unit of the second amount of information units coincides with a codeword border. Thus, a clear separation between the first amount of information units and the second amount of information units and the correspondingly associated error protection procedures on the transmitter side on the one hand and the receiver side on the other hand is obtained.

The audio transmitter processor additionally comprises the error protection coder 508 for processing the predefined number of codewords individually to obtain a plurality of processed codewords representing the error protected frame. For this procedure, the error protection coder may comprise an entropy-adding or block encoder such as a Reed-Solomon encoder. Alternatively or additionally, the error protection coder may comprise another non-codeword individually processing device such as a CRC or Hash value processor that processes one or more of the codewords of the first predefined subset to obtain a first processing result such as a first Hash value or to process one or more of the codewords of the second predefined subset to obtain a second processing result such as a second Hash value and for adding the processing results or one of the first and the second processing results to the predefined number of codewords to obtain the plurality of processed codewords. The first processing result is only derived from the first predefined subset and the second processing result is only derived from the second predefined subset. Preferably, the order of error protection processing is so that the first and the second processing results are calculated from the codewords that have not yet be subjected to any error redundancy processing such as the codewords at an input into the Reed-Solomon or any other error protection block encoder. It is preferred that the first and the second processing results are added to the first and second predefined sets of codewords either in a separate additional codeword or to an empty space that is still available within either the first or the second set and the codewords and the processing results such as the Hash values are protected by means of the Reed-Solomon encoder processor.

The error protected frame obtained by the error protection coder 508 of Fig. 5 is forwarded to a preferably wireless transmitter such as a DECT (digital enhanced cordless telephone) standard conformant device. This transmitter 510, therefore, sends the error protected frame out into the (wireless) error-prone channel.

Preferably, the frame builder 506 is configured to determine the border between the first amount of information units and the second amount of information units based on an information capacity of the second predefined subset so that the information capacity of the second predefined subset is equal to the amount of information units of the second amount of information units. As soon as the frame writer 504 has written an amount of data equal to the whole amount of data for the whole frame less the capacity of the second predefined subset, the first amount of data of the audio frame is complete and the

second amount of data starts at the start of the first codeword of the second predefined subset of the codewords.

The audio encoder 500 or source encoder for generating the first and second amounts of information units relies on a frame-wise processing and using a predefined time portion of an audio signal. The information units comprise a set of obligatory information units such as the data generated by an arithmetic encoder or, generally, variable length encoder. The information units have a variable number of residual information units that provide an improved quality representation of the time portion of the audio signal and this data is written into the bitstream, when the bit budget is not yet completed by the arithmetic encoder. The source encoder 500 is configured for using a variable length coding rule resulting in the signal-dependent number of information units for the predefined time portion. The frame builder is configured to build the encoded audio frame so that the encoded audio frame has a fixed size in encoded information units, and the audio encoder 500 is configured to determine the variable number of the residual information units as a difference between the predefined frame size and the number of obligatory bits.

Preferably, the frame builder 504 is configured to determine the border between the first amount of information units and the second amount of information units signal-adaptively from frame to frame, so that, depending on the audio signal for a frame, the border represents a border information unit related to a different audio information of the frame or being interpreted differently by an audio decoder. Preferably, the border information unit refers and represents a certain spectral value or a certain combination of spectral values in a spectral domain audio coder as is preferably used within the audio encoder 500 and which will later be discussed with respect to Fig. 21 for the encoder and Fig. 22 for the decoder.

In a preferred implementation of the present invention, the frame writer 504 or, generally, the frame builder 506 is configured to write the first amount of information units into the first predefined subset identified at 600 in Fig. 6. In step 601, the border between the first and the second amount of information units is determined and as soon as the first subset is filled with information units. As indicated in item 602, the second amount of information units is written into the second predefined subset of the codewords.

Preferably, as illustrated in Fig. 7, the first predefined subset of the codewords has an order of codewords and specified reference positions. The same is true for the second

predefined subset of the codewords that also preferably has an order of codewords and reference positions.

Fig. 9a illustrates a first possibility of arranging a predefined subset of the codewords within a frame of codewords. In the Fig. 9a embodiment, the first subset of the codewords are codewords 4, 5, 6, 7 and the second subset of the codewords are codewords 1, 2, 3. The codewords of the second subset of the codewords – that receive the second amount of information units that are, preferably, the psychoacoustically less important audio data – are all positioned adjacent to each other and at the frame start position. The codewords of the first subset of the codewords – that receive the first amount of information units that are, preferably, the psychoacoustically more important audio data – are all positioned adjacent to each other and at the frame end position.

The first predefined subset of the codewords is predefined by the fourth codeword in the frame and by an order or sequence of codewords from codeword 4 to codeword 5, from codeword 5 to codeword 6, and from codeword 6 to codeword 7. The first predefined subset identifies the codewords and the order of the codewords for the writing direction. The frame builder is configured to write, in the preferred embodiment, the first subset of the codewords, i.e., the codewords 4, 5, 6, 7 as indicated by the arrows that start at reference positions of the codewords. The writing operation from left to right starts at the start of the fourth codeword as the reference position and the writing in the opposite direction starts at the end of the seventh codeword as the reference position, i.e., at the frame end position. The second predefined subset also identifies the codewords and the order of the codewords for the writing direction correspondingly.

The second subset of the codewords is predefined by the first codeword in the frame and by an order or sequence from the first codeword to the second codeword and from the second codeword to the third codeword. In case the codewords or a subset are all adjacent to each other, the order or sequence information is implicitly given by the writing or reading direction. Again, the writing in the left direction from left to right is at the frame start position of the first codeword and the writing from the right end of the second subset, i.e., from the codeword 3 starts from the end of the third codeword in the direction to the frame start position.

Naturally, the number of codewords in the first subset and in the second subset is freely selectable and, the higher the number of the codewords of the second subset is, the lower

is the necessity for a full frame loss concealment. However, it has to be made sure that the number of codewords of the first subset is large enough so that a partial frame loss concealment with an acceptable quality can be done when all the codewords of the second subset or, for example, the codewords 1 and 3 in the Fig. 9a embodiment of the second subset are erroneous as can be detected by the audio receiver processor illustrated in Fig. 13.

Fig. 9b illustrates an alternative implementation of the first predefined subset and the second predefined subset. Once again, both subsets define codewords that are adjacent to each other, but the first predefined subset is now aligned with the frame start position and the second predefined subset of the codewords is now aligned with the frame end position.

Fig. 9c illustrates another alternative, where the first subset and the second subset are arranged non-adjacent to each other, i.e., a codeword of the second subset, i.e., codeword 2 is interspersed between two codewords, i.e., codeword 1 and codeword 3 of the first predefined subset. Fig. 9c once again indicates the writing direction for the individual codewords, and it becomes clear that, for example, codeword number 5 is written from both sides, and when this is the case, the writing pointers for writing the second subset of the codewords will meet each other at some place within codeword number 5.

In the Figs. 9a to 9c embodiments, the arrows above the codeword representation indicate the direction of writing when the preferred implementation of two writing pointers is applied that is subsequently discussed with respect to Fig. 8. Particularly, as discussed with respect to Figs. 9a to 9c, the frame builder 5 is configured to use, as the second subset of the codewords a predefined number of adjacent codewords at one side of the frame of the encoded audio data as, for example, illustrated in Fig. 9a, or to use, as the first subset of the codewords, a first predefined number of adjacent codewords at another side of the frame of the encoded audio data as, for example, illustrated with respect to Fig. 9a or 9b, where a sum of the first predefined number of codewords and the second predefined number of codewords is equal to the total predefined number of codewords. Alternatively, as illustrated in Fig. 9c, at least one codeword of the first subset is located between two codewords of the second subset or vice versa.

Preferably, the frame builder 506 is configured to write the first amount of information units into the first predefined subset using a reference position of the first subset of the codewords and, as soon as the first predefined subset is filled, the second amount of information units is written at reference positions of the second predefined subset, and the frame builder 506 is configured to determine the border as the last information unit written into the last codeword of the first predefined subset or as the first information unit written at a reference position of a first codeword of the second predefined subset. Preferably, the first and the second amounts of information units are selected in such a way that all or at least a majority of the information units of the first amount of information units is psychoacoustically more important than a majority or all the information units of the second amount of information units.

Alternatively or additionally, and as discussed with respect to the transmitter side, only partial frame loss concealment is envisioned in an audio receiver processor when only information units in the second amount of information units are detected as corrupted and wherein a full frame loss concealment is envisioned in the audio receiver processor, when information units in the first amount of information units are determined as corrupted.

As is discussed later on with respect to Fig. 1 or 2, the encoded information units are from at least two categories that are selected from a group of categories consisting of fixed length side information, variable length side information, temporal noise shaping information, one or more most significant bits of a first frequency portion of a spectrum, one or more most significant bits of a second frequency portion of the spectrum, wherein the second frequency portion is higher than the first frequency portion, one or more least significant bits or sign bits of the first frequency portion, one or more least significant bits or sign bits of the second frequency portion and residual bits, wherein, if generated by the audio encoder, the fixed length side information, the variable length side information, the temporal noise shaping information, the one or more most significant bits of the spectrum of a first frequency portion and the one or more least significant bits or sign bits of the first frequency portion are used as categories for the first amount of information units, and wherein the most significant bits of the second frequency portion, the one or more least significant bits or sign information units of the second frequency portion or the residual bits are used as categories for the second amount of information units.

In a preferred embodiment illustrated in Fig. 8, two writing pointers 810, 812 are used. The first writing pointer 810 is configured to operate and write in a first writing direction, and

the second writing pointer 812 is configured to operate and write in a second writing direction, which is opposite to the first writing direction. The data for the first writing pointer is obtained by a first controlled input and is taken, for example, from an input buffer 802 in which any imaginable form of audio encoder output data is or in which a specified intermediate format such as a standardized format, as for example, discussed with respect to Fig. 1 for the LC3 (Low Complexity Communication Codec) audio encoder is present.

In a first example case, the data in the input buffer comes directly from an encoder. In this case, the data are taken as they come from the encoder. In an example of this first example case, the encoder writes LSBs and Signs for a spectral line or a spectral line tuple in the first controlled input 804 and MSBs for this same spectral line or spectral line tuple in the second controlled input 806.

In a second example case, the data stem from an already written frame. Then, the controller applies a bitstream or frame parser reading the bitstream or frame and providing the data to the controlled inputs in the read/parsed order. In an example of this second example case, the parser reads LSBs and Signs for a spectral line or a spectral line tuple and provides this data into the first controlled input 804 and the parser reads MSBs for this same spectral line or spectral line tuple and provides this data into the second controlled input 806.

There is a second controlled input 806 that also accesses the input buffer 802 and that provides data to the second writing pointer 812 that is configured to write in the second (opposite) direction. The controller 800 is configured to control at least the first and second writing pointers 810, 812 and preferably additionally the inputs 804, 806. The controller receives, as an input, the number of codewords of the second set or, correspondingly, the number of less important information units, i.e., the capacity of the codewords of the second predefined subset of the codewords. The controller preferably has stored information about the predefined first and second subsets and the associated orders, i.e. information on the codeword numbers/positions in the frame and/or the order of the codewords for a respective subset.

The controller 800 controls the inputs 804, 806. The controller additionally sets the first and second pointers to the start positions/addresses in a frame for the first amount of

information units. The pointer 810 is incremented and the pointer 812 is synchronously decremented.

The controller 800 detects that all codewords of the first subset are written into the output buffer and the controller sets the first and second pointers' start positions/addresses for the second amount of information units and, subsequently synchronously increments/decrements the writing pointers in order to additionally performing writing of the second amount of information units. Preferably, the order of the procedures done by the controller 800 is as indicated in Fig. 8 but different orders can be performed as well. However, determining the start of the second amount of information units based on when the first amount of information units is written is an automatic and low efficiency and low complicated way of determining the first and second amount of information units from frame to frame even though the audio encoder is operating in an audio signal dependent way.

There exist several applications for the frame builder as is discussed with respect to Figs. 10 and 11. Particularly, Fig. 10 illustrates the procedure, when a direct writing is performed. In step 100, the frame builder receives the encoded audio data from the audio encoder and determines the first predefined subset of the codewords. It is written from both sides and the writing is stopped when the writing pointers meet each other as indicated at item 101. Writing is continued into the second predefined subset of the codewords until all information units are written. Particularly, as indicated at 102, the writing is continued at the start or the end of the frame for the second predefined subset at least with respect to one writing pointer.

The frame builder can also be used for rearranging an already existing bitstream such as an LC3 bitstream format. In this procedure, an encoded audio frame exists in a specified/standardized intermediate format where, for example, all codewords for the first set or all codewords of the second set are located adjacent to each other at the left or the right side of the finally to be rearranged frame.

In step 200, the first predefined subset is determined. In step 202 the intermediate format is read and parsed from both sides. In step 204 reading and parsing is stopped, when the capacity of information units in the first predefine subset is read. In step 206, the frame is written by exchanging the data up to the left pointer with the data between the pointers in the specific embodiment that is, for example, illustrated in Fig. 3 to be discussed in detail

later. A writing operation in the Fig. 11 embodiment is only done in step 206, since steps 202 to 204 only refer to reading and parsing and doing other procedures. The writing does not necessarily have to be an incremented or decremented writing as in the direct writing discussed with respect to Fig. 10, but the writing is done by exchanging certain contiguous amounts of information units.

Fig. 12 illustrates a preferred implementation of the error protection coder 508 illustrated in Fig. 5. In a first step 210, the error protection processor 508 calculates a first Hash value from the first predefined subset of the codewords only, without any codeword from the second subset of the codewords. In step 212, a second Hash value is calculated from the second predefine subset only, i.e., without any codeword from the first predefined subset.

In step 214, the first and the second Hash values are added to the subsets of the frame as indicated at 214. In step 216, a block coding such as a Reed-Solomon-coding is performed to the subsets and the Hash values in order to obtain the error protected frame that is forwarded to the preferably wireless transmitter 510 that is, for example, a DECT conformant transmitter.

Fig. 21 illustrates a typical audio encoder or source encoder such as the audio encoder 500, but the audio encoder 500 can also be implemented by any other audio encoder that operates in a variable-length way, i.e., generates a certain amount of bits for a time portion of an audio signal that varies from frame to frame when the quality is kept the same.

An audio signal is input into an analysis windower 52 that, preferably, operates in an overlap way, i.e., has a time advance value that is lower than the time length of the window. This data is a (windowed) time portion for a frame and is input into a time-frequency transform 54 that is preferably implemented as an MDCT (modified discrete cosine transform).

In block 56, an optional temporal noise shaping operation is performed which consists of a prediction over frequency. The output of the TNS stage 56 are prediction residual values and, additionally, TNS side information is output preferably into the entropy coder 60 that is a variable length entropy coder such as an arithmetic coder. The MDCT output spectral values or the TNS spectral residual values are quantized by a quantizer 58 that may or

may not be controlled in a psychoacoustic way and the quantized data is input to the variable length entropy coder such as an arithmetic coder. The audio encoder additionally comprises a residual coder for generating residual bits that are necessary in order to fill the frame with additional bits, when the variable length entropy coder does not fully consume the available bit budget. Other features that can be used or not are noise filling, global gain application or spectral noise shaping. At the output of the block 60, a bit stream multiplexer is arranged receiving data from block 60 that can be MSBs, LSBs and Signs and other data.

On the decoder-side illustrated in Fig. 22, a variable length entropy decoder is there, which is, once again, an arithmetic decoder, for example. The result of the variable length entropy decoder is input into a dequantizer 74 and the output of the dequantizer 74 is processed by an inverse TNS processor when TNS processing is available or, when TNS is not performed, the output of the dequantizer 74 is forwarded to the inverse frequency-time transform that can, for example, be an inverse MDCT transform as indicated at 78 in Fig. 22. The output of block 78 is forwarded to the synthesis windower and overlap/add processor 80 that, finally, obtains a time domain decoded audio signal. The TNS data used by the inverse TNS processor 74 is typically derived from the bitstream and is even preferably derived from the variable length entropy decoder, but other ways in order to encode and process and transmit data can be used as well.

Subsequently, a preferred implementation of the present invention is discussed with respect to Figs. 1 to 4. This embodiment relies on the order and arrangement of the first and second predefined subsets of codewords illustrated in Fig. 9a, but is equally applicable to other arrangements as well. The subsequent embodiment gives a clear detail up to a bit level but, of course, the specific figures are embodiments only and it is clear that other detailed figures/numbers can be used as well.

The focus within this section is exemplarily on the rearrangement scenario to align the data to a given error protection and detection scheme, allowing for individual frame loss concealment procedures depending on the availability of the psychoacoustically more important data and the psychoacoustically less important data.

The preferred embodiment is explained based on the LC3 bitstream payload. This bitstream payload can roughly be subdivided into the following categories (see before):

1. Side information, fix part (solid red) and variable part (red-white striped diagonally)
2. TNS, arithmetically encoded (TNS_{arith})
3. MSBs of the spectrum, arithmetically encoded (MSB $Spec_{arith}$)
4. LSBs and Signs of the spectrum (LSB-Sign Spec)
5. Residual

Note that the spectrum is encoded by means of spectral tuples, whereas each tuple represents two subsequent spectral lines.

The arrows indicate the writing and reading direction. The writing / reading order is as follows:

1. The fix part of the side information is written from right to left.
2. The variable part of the side information is written from right to left.
3. The arithmetically coded TNS data is written from left to right.
4. The arithmetically coded MSBs of the spectrum as well as the LSBs and the Signs of the spectrum are written synchronously (spectral line by spectral line, starting with the spectral line representing the lowest frequency), whereas the arithmetically coded MSBs are written from left to right and the LSBs and Signs (being not arithmetically coded) are written from right to left).
5. If there are still bits left between those two data parts, Residual bits are written from right to left.

While categories 1 and 2 are always treated as psychoacoustically important, categories 3 to 5 are generally treated as less psychoacoustically important. However, the relevancy is not constant for all those bits within this region:

- Bits representing the residual are least significant;
- Bits representing spectral tuples are the psychoacoustically less important, the higher the frequencies the covered spectral lines represent, i.e.
  - Bits representing spectral tuples covering spectral lines with a higher frequency are less significant;
  - Bits representing spectral tuples covering spectral lines with a lower frequency are more significant.

Fig. 1 and Fig. 2 show the distribution of bits based on their psychoacoustic relevancy. Red indicates a high relevancy; green indicates a small relevancy. There are two transition phases, and both of them change dynamically on a frame-by-frame basis.

The last written bits in the bitstream are the psychoacoustically least important bits. Their location is variable. It can be determined directly while writing or reading (and parsing) the bitstream by checking until the given number of bits to be treated as psychoacoustically less important is left. This given number covers the residual and the bits of a certain number of spectral lines (MSBs as well as LSBs and Signs). The coverage of the spectrum starts with the lines representing the highest frequencies. The higher the given number of less important bits, the lower is the upper frequency, which can successfully be decoded if those less important bits are distorted.

The number of codewords to be treated as psychoacoustically less important N_{pccw}, and the amount of bits being treated as psychoacoustically less important, is a tuning parameter. This tuning parameter is determined preferably from a characteristic of the error protection code. For one embodiment in LC3, the optimal size has been derived experimentally depending on the protection strength (error protection mode m) and the slot size N_s as examples for a characteristic of the error protection code, which specifies the size of the channel encoded frame, i.e. the gross frame size, in bytes. This is embodied in the following formula:

$$N_{pccw} = \begin{cases} \lfloor 0.080447761194030 \cdot N_s - 1.791044776119394 + 0.5 \rfloor, & \text{for } m = 3 \text{ and } N_s \geq 80 \\ \lfloor 0.066492537313433 \cdot N_s - 1.970149253731338 + 0.5 \rfloor, & \text{for } m = 4 \text{ and } N_s \geq 80 \\ 0, & \text{otherwise} \end{cases}$$

Fig. 3 gives an example of an audio frame before and after the rearrangement as well as the assignment to the codewords provided by the channel codec. It also shows the parsing of the rearranged bitstream payload on decoder side.

The gross bitrate in this example is 76800 bits/s at 10ms framing, resulting in 96 bytes per frame. For this frame length, the channel codec provides 13 codewords: 3 codewords with a gross size of 7 bytes and 10 codewords with a gross size of 7.5 bytes. With error protection mode 4 (=3 bytes protection overhead), the fec (forward error correction) overhead is 39 bytes, leaving 57 bytes for the payload, split over 3 codewords with a net size of 4 bytes and 10 codewords with a net size of 4.5 bytes.

Fig. 3A shows the bitstream payload of one frame separated into 57 byte packets, resulting in 456 total bits [0:455]. The red block corresponds to the static side information, whereas the red/white shaped region corresponds to the dynamic side information, which may vary from frame to frame depending on the coded data.

The bits treated as less significant are shown in blue, delimited by the bit borders b_left and b_right (in the given example, b_left=184, b_right=315). This area overlaps the residual bits and additionally covers bits from the "MSB $Spec_{arith}$" and from the "LSB+Sign Spec", starting from the highest frequencies. The number of bits from the "MSB $Spec_{arith}$" is usually higher than the number of bits from the "LSB+Sign Spec", since usually more bits per spectral line are consumed to encode the MSBs than to encode the LSBs and Sign.

"MSB $Spec_{arith}$" up to b_left (written from left to right, shown in green) and "LSB+Sign Spec" up to b_right (written from right to left, shown in white) jointly represent the spectral tuples from zero Hertz up to the highest frequency encoded up to this point. If one more spectral tuple should be considered as psychoacoustically less important, at least one border would move outwards; if one more spectral tuple should be considered as psychoacoustically more important, at least one border would move inwards.

Fig. 3B shows the frame after the rearrangement: The blue part [184:315] is exchanged with the green part [0:183]. Note, that the blue part may be smaller, of equal size, or larger than the green part.

Fig. 3C displays the payload of the 13 codewords as input into the channel codec.

Fig. 3D shows the received bitstream payload. It exemplarily shows two distorted codewords.

Fig. 3E shows the decoding process. It exemplarily shows the distorted bits encapsulated between the two bit borders be_bp_left and be_bp_right. Frequency bins of spectral tuples represented by the bits within this range should be synthesized by the partial frame loss concealment.

Obviously, the assignment of the bits within the bitstream payload to the two envisioned classes does not directly map to the codewords provided by the channel coder.

In the given example, the part between b_left and b_right in Fig. 3A [184:315] is assigned to the second class, whereas the other bits [0:183] and [316:455] are assigned to the first class. Here, the bits assigned to the second class [184:315] in Fig. 3A do not fit into the codeword structure in Fig. 3C without increasing the number of codewords for the second class. The second class would lie partially in the codeword 6 [184:203], fully in the codewords 7 to 9 [204:311] and partially in the codeword 10 [312:315]. Hence, a rearrangement is required as shown in Fig. 3B: Now the second class [0:131] fits perfectly into the first four codewords 1 to 4.

In the given example, 4 codewords belong to the second class and 9 codewords belong to the first class. The number of bits to be stored in either class is such limited by the amount of payload bits offered by the codewords of this class. In the given example, codewords 1 to 3 provide each a payload of 4 bytes, whereas codewords 4 to 13 provide each a payload of 4.5 bytes. This results in

- 3*4 bytes+1*4.5 byte=16.5 byte for the psychoacoustically less important bits and
- 9*4.5 bytes=40.5 byte for the psychoacoustically more important bits.

While the number of psychoacoustically less important bits (block_size) is predetermined, the location of the borders (b_left and b_right) varies on a frame-by-frame basis. After the rearrangement, those bits are always located at the same spot, which enables an efficient channel coding.

At encoder side, the bitstream bs_enc is rearranged as follows:

$$\text{bs_rearranged(k)} = \begin{cases} \text{bs_enc(b_left} + k), & 0 \le k < \text{block_size} \\ \text{bs_enc}(k - \text{block_size}), & \text{block_size} \le k < \text{b_left+block_size} \\ \text{bs_enc}(k), & \text{b_left+block_size} \le \text{k} < \text{len} \end{cases}$$

Where len is the net size of the frame in bits and block_size is the number of less significant bits.

On decoder side, the border b_right, which is the border where the two pointers reading "1. MSB $\text{Spec}_{\text{arith}}$" and "1. LSB+Sign Spec" meet each other (see Fig. 3E), is determined while decoding the bitstream. The number of psychoacoustically less important bits is known from the forward error protection configuration (error protection mode and number of codewords N_{pccw}). The rearrangement of the frame bs_rearranged at decoder side is done as follows:

$$\text{bs_dec(k)} = \begin{cases} \text{bs_rearranged (block_size} + k), & 0 \leq k < \text{b_left} \\ \text{bs_rearranged } (k - \text{b_left}), & \text{b_left} \leq k < \text{b_left+block_size} \\ \text{bs_rearranged}(k), & \text{b_left+block_size} \leq \text{k} < \text{len} \end{cases}$$

If no bit-errors are applied on the bs_rearranged frame, bs_enc is equal to bs_dec.

As indicated before, the rearrangement can be done either as a post-processing step, or directly during the writing / reading process.

The following parameters are static:

1. The slot size N_s specifies the size of the channel encoded frame in octets (bytes). In the given example $N_s = 96$.

2. N_{cw} specifies the number of codewords that are used to encode the data frame

$$N_{cw} = \left\lceil \frac{2N_s}{15} \right\rceil$$

In the given example $N_{cw} = 13$.

3. The parameter L_i, which is defined for $i = 0..N_{cw} - 1$, specifies the length of the i[th] codeword in semi-octets (i.e. nibbles) and is given by:

$$L_{13-i} = \left\lfloor \frac{2N_s - i - 1}{N_{cw}} \right\rfloor + 1$$

In the given example $L_{1..3} = 14$, $L_{4..13} = 15$. Note that the enumeration is done differently in Fig. 3C.

4. The parameter $d_{i,m}$, which specifies the Hamming distance of (RS)-code i in error protection mode m, is given by:

$$d_{i,m} := 2m - 1 \quad \text{for } i = 0..N_{cw} - 1$$

where $m > 1$. In the given example $d_{i,m} = 2 * 4 - 1 = 7$

5. The number of codewords assigned for the psychoacoustically less important bits is derived based on the frame length and the error protection mode (see above). In the given example $N_{pccw} = \lfloor 4.913134 \rfloor = 4$

6. The size of the partial concealment block in semi-octets can be derived as

$$N_{pc} = \sum_{i=N_{cw}-N_{pccw}}^{N_{cw}-1} L_i - d_{i,m} + 1$$

In the given example $N_{pc} = 33$.

7. The number of less important bits (block_size) can be derived as:

$$\text{block_size} = 4N_{pc}$$

In the given example $\text{block_size} = 132$.

8. Accordingly, the starting point for writing the TNS data is known (16.5 bytes from the left).

The following needs to be done for each frame on encoder side:

1. Write fixed part of the side information, starting from the right end of the bitstream payload, from right to left.

2. Write variable part of the side information, starting at the left end of the fixed part of the side information, from right to left

3. Write TNS data, starting block_size from the left end of the bitstream payload, from left to right.

4. Write MSBs "1. MSB $Spec_{arith}$" of the spectral data, starting from the right end of the TNS data, from left to right, up to b_left+block_size-1=b_right; and write LSBs and signs "1. LSB+Sign Spec" of the spectral data, starting from the left end of the side info, from right to left, up to b_left+block_size. Note, that b_left and b_right are not known in advance.

5. The border b_left+block_size-1=b_right is determined, when the two pointers reading "1. MSB $Spec_{arith}$" and "1. LSB+Sign Spec" meet each other (see arrows in Fig. 3E).

6. Continue writing MSBs "2. MSB $Spec_{arith}$" of the spectral data, starting from the left border of the bitstream payload, from left to right; and continue writing LSBs and signs "2. LSB+Sign Spec" of the spectral data, starting from block_size - 1, from right to left.

7. Write residual, starting from the left end of the LSBs and signs of the spectral data, from right to left.

The reading on decoder side can be done similar to the described writing on the encoder side.

Fig. 3 illustrates this process of writing or reading the rearranged bitstream payload. Fig. 3A shows the 'normal' bitstream payload, whereas Fig. 3B shows the rearranged bitstream payload. As outlined above, this rearranged bitstream payload can immediately be written or read. Alternatively, a rearrangement can also be performed as a post-process on encoder side or as a pre-process on the decoder side. However, the bitstream needs to be parsed for this rearrangement process as follows:

- Original order → rearranged order: b_left and b_right need to be found by counting the remaining bits to be written – the borders are reached, if this number equals block_size.

- Rearranged order → original order: b_right needs to be found by observing the pointers reading "1. MSB $Spec_{arith}$" and "1. LSB+Sign Spec" – the border is reached, when both pointers meet each other.

While the range of bits representing the psychoacoustically less important bits is predetermined by the channel codec, b_left and b_right could be on an arbitrary bit position.

In one preferred embodiment, the arithmetic encoder/decoder operates on byte granularity. In this example, b_left already falls to a byte boundary. This is reflected in Fig. 3A, where b_left=184.

Since, however, the underlying channel coder (Fig. 3C) operates on nibble (4 bits) granularity, block_size might not be a multiple of bytes and thus b_right might also not fall on a byte boundary. In this case, a mismatch will occur after the rearrangement as described above. In the given example, such mismatch is visible in byte 17 [128:135] (Fig. 3E), where the arithmetic decoder would have to start decoding at bit position 132, which is not a byte boundary.

To cope with this, the block_size is now derived as:

$$\text{block_size} = 8\left\lceil \frac{N_{pc}}{2} \right\rceil$$

Fig. 4 shows an example on this embodiment, considering the same parameters as given in Fig. 3. Here, $\text{block_size} = 136$ bits, i.e. b_right is now also on a byte boundary, leading to the starting point of the arithmetic decoder at the bit position 136, resulting at the integer byte position 17 (Fig. 4E, the arithmetic decoder starts with decoding TNS_{arith}). The right nibble of byte 17 [132:135] (displayed striped in blue-white) now contains bits being treated as psychoacoustically less important, although it is assigned to a codeword intended to cover psychoacoustically more important data. This has the following effects:

- If there were uncorrectable bit errors in codeword 5, the whole frame would be synthesized with full frame loss concealment even if the errors only affect the right nibble of byte 17, i.e. bits [132:135].

- If the codewords 5 to 13 are good and there are uncorrectable bit errors in codeword 4, which may be just located in the left nibble of byte 17, i.e. bits [128:132]:, then the right nibble of byte 17 can still be decoded, as it is coded in the good codeword 5.

It is noted, that

- b_left might shift slightly to the next left byte boundary, if it is not already on a byte boundary like in the example given in Fig. 3.
- The number of spectral tuples belonging to the psychoacoustically less important data might slightly increase, whereas the number of spectral tuples belonging to the psychoacoustically more important data might correspondingly slightly decrease. Therefore, in some impaired situations, it can happen that the number of frequency tuples that can be decoded is less than in the nibble/bit granularity case. However, quality wise this will have a marginal influence, but allows for a more efficient implementation of the arithmetic encoder/decoder.

Although the above example addresses the situation, where the granularities are in an integer relation to each other, an analogous processing is used in case of non-integer relations of the granularities or in case of the arithmetic encoder granularity being lower than the error protection granularity.

A feature of the preferred approach for a bitstream rearrangement (separation of psychoacoustically less important from psychoacoustically more important bits) is, that the bits are written or read chronologically – but not spatially – by the audio encoder/decoder, starting with the high important bits and ending with the low important bits, since two bitstream writers write simultaneously into the bitstream, or two bitstream readers read simultaneously out of the bitstream (in the present case in different directions), and that – due to the variable length coding – no a-priori information is given, where those two writing or reading pointers are located, when a certain amount of data is written or read. Such locations are just known once this amount of data is written, or once this amount of data is parsed (i.e. read and evaluated) – either, because a certain amount of data is written or read, or because the two pointers have met each other during writing or reading.

Although a rearrangement as a post- or preprocess is possible after the encoding step or prior to the decoding step, a direct writing or reading of the rearranged bitstream is preferable, since the rearrangement requires an additional parsing step.

Subsequently, preferred embodiments of the present invention that are related to the decoder or receiver side are discussed with respect to Figs. 13 to 20.

Fig. 13 illustrates an audio receiver processor for processing a received error protected frame. The audio receiver processor comprises a receiver interface 300 for receiving the error protected frame and for outputting the error protected frame to an error protection processor 302. The error protection processor 302 is configured for processing the error protected frame to obtain an encoded audio frame. Particularly, the error protection processor receives information on the first or the second predefined subset and preferably, only from the first predefined subset and, even only on the codewords belonging to the first predefined subset. This information is sufficient so that the error protection processor 302 can check whether a predefined subset of the codewords of the encoded audio frame derived from the error protection processor 302 comprises an error.

The error protection processor 302 processes the error protected frame, outputs a frame with codewords and additionally generates an information whether the frame with codewords that is typically a fixed length frame has an error within the first predefined subset of the codewords.

In this context, it is to be mentioned that the codewords input into the error protection coder 508 of the audio transmitter processor of Fig. 5 or the codewords output by the error protection processor 302 of the audio receiver processor can also be considered to be payload codewords, and that the codewords output by the error protection processor 302 of the audio receiver processor or the codewords input into the error protection coder 508 of the audio transmitter processor of Fig. 5 are termed just codewords.

The audio receiver processor comprises an error concealer or error concealment indicator 304 configured to perform a frame loss concealment operation in case of a detected error in the first predefined subset of the codewords or to generate an indication for such an error concealment operation to be performed at e.g. a remote place.

The audio receiver processor comprises a frame reader 306 and a subsequently connected audio decoder 308. The frame reader 306 can be controlled by the error protection processor 302 or the error concealer or the error concealment indicator 304. Particularly, in case the error protection processor 302 determines an error in the first predefined subset of the codewords which results in a typically full frame loss concealment, the frame reader 306 and the audio decoder 308 may be controlled in such a way that these blocks do not have to operate for this frame anymore.

In case the error protection processor 302 determines an error in the second predefined subset of the codewords but not in the first predefined subset of the codewords, the frame reader 306 is controlled to read in the first predefined subset of the codewords but a reading of the second predefined subset of the codewords can be skipped, since an error has been detected or at least a reading of a certain codeword in which an error has been detected can be skipped if not all codewords of the second predefined subset of the codewords are erroneous.

In case the error protection processor 302 has determined that both subsets of codewords are error free, i.e., do not contain any error that has not been corrected by the error protection processor procedure, the frame reader 306 is configured to read the encoded audio data in the first and second predefined subsets using predefined information on the first and second predefined subsets and the frame reader 306 may output the read audio data in any form or a specified intermediate format for the purpose of processing by the audio decoder 308.

When the error concealer or the error concealment indicator 304 has been controlled by the error protection processor 302 to perform a partial frame loss concealment, the error concealer may generate synthesis audio data and forward the synthesis audio data to the audio decoder 308 so that the audio decoder can use this concealment data such as higher spectral values or residual data instead of the transmitted but error-affected audio data in the second set of second codewords. Depending on the implementation, the error concealer or the error concealment indicator 304 or frame loss concealer in Fig. 13 uses data from one or more earlier frames, and the concealer 304 can be integrated into the audio decoder so that both functionalities are integrated into each other. In case of the error concealment indicator, the concealment is done at a place remote from the error protection processor, and the audio decoder receives an indication to perform the concealment from device 304.

Preferably, the error protected frame has two stages of error protection. The first stage of error protection is the redundancy introduced by the block coder such as the Reed-Solomon encoder on the transmitter side. The further and second line of protection is the calculation of one or more Hash codes over the first predefined subset of the codewords on the one hand and the second predefined subset of the codewords on the other hand.

Although the error protection processor and, particularly, the block code such as the Reed-Solomon code applied by the error protection processor can detect and correct several errors, the case may be that some errors survive the Reed-Solomon decoding without any detection and without any correction or that the error correction has "corrected" errors in the wrong direction. In order to find these errors as well, a Hash verification is performed using a transmitted Hash output by the Reed-Solomon decoding operation and a comparison of this transmitted Hash value with a Hash value derived from the decoded first (or second) predefined subsets of codewords.

A preferred implementation is illustrated in Fig. 14. In step 400, a Reed-Solomon decoding with an error detection/correction is performed. This procedure results in a decoded frame with codewords and transmitted first and second result values that are preferably implemented as Hash values. In step 402, a first Hash value is calculated from the first predefined subset and in step 404, the calculated first Hash value is compared to the transmitted first Hash value. In case both Hash values are equal, the procedure goes on to step 406. However, in case both Hash values are not equal, an error has been detected and the processing is stopped and a full frame loss concealment is started as indicated in step 408.

However, when it has been determined that the first Hash value and the transmitted first Hash value are equal to each other, step 406 is performed in which the second Hash value is calculated and compared to the transmitted second Hash value. In case both Hash values are not equal, than the procedure indicated in step 410 is applied, i.e., a partial frame loss concealment is performed. When, however, it is determined that both Hash values with respect to the second predefined subset of the codewords are equal, the frame reader and the audio decoder are controlled to perform an error-free decoding operation. The procedure illustrated in Fig. 14 is preferably implemented by the error concealer or error concealment indicator 304 and/or by the error protection processor 302.

Fig. 15 illustrates a preferred implementation of partial frame loss concealment illustrated in block 410 of Fig. 14. In step 420, it is determined whether only one or more codewords from all the codewords of the second predefined subset of the codewords are erroneous. To this end, an information is received from, for example, the block decoder such as the Reed-Solomon decoder or a CRC check per codeword. Typically, the Reed-Solomon decoder will indicate that or which codeword of the second predefined subset of the codewords is erroneous. It is, for example, indicated by block 420 that only one or two

codewords are erroneous and other codewords of the second set of predefined codewords are correct. The data of these other non-corrupted codewords is used as much as possible for the normal decoding or the partial concealment. In step 422, the one or more non-erroneous blocks or codewords of the second subset are read.

In step 424 it is determined, which meaning such as the spectral range or the residual data is reflected by the one or more erroneous codewords. To this end, an information on codeword order and reference positions of the second predefined subset of the codewords is useful. Step 424 determines the meaning of the erroneous codewords so that step 426 can synthesize concealment data for the erroneous blocks such as the spectral range determined by block 424. Alternatively, a kind of error concealment may also be that residual data that have been indicated as corrupted are simply skipped so that the residual decoding and the corresponding quality improvement that would be obtained in case of non-erroneous codewords is simply not performed as a kind of an error concealment procedure in a non-problematic situation where only the residual data are corrupted.

However, in case of a determination that a certain spectral range is corrupted, concealment data for this spectral range is generated by block 426.

In block 428 the read data from the first predefined subset and the correct data from codewords of the second predefined subset and the concealment data are combined and decoded in order to finally obtain the decoded audio signal for the time portion (frame) of the audio signal resulting from a partial frame loss concealment procedure.

Fig. 16 illustrates a general representation of the procedure performed by block 424. By parsing the bitstream with the erroneous data, the meaning of the data with respect to which spectral values these data represent is determined in order to obtain the concealment spectral range. However, the actual values of these data are not used since they have been determined to be erroneous. The result of the procedure subsequent to steps 424 and 426 will be that non-erroneous spectral range data are obtained until the concealment border and erroneous spectral data that are replaced by synthesis/concealment data exist for the spectral range between the concealment border and the maximum frequency. However, in other embodiments, the case may be that the concealment spectral range does not fully extend to the maximum border but only covers a certain spectral range between the concealment border and another border which is

lower than the maximum frequency required by the audio decoder. The audio decoder still receives correctly received data between the concealment border and the maximum frequency.

In an embodiment, the error concealer or error concealment indicator 304 is configured to generate substitute data, and this data is, then, decoded or, in general, used by the decoder together with the non-erroneous data. In another embodiment, the error concealer or error concealment indicator 304 only generates an error concealment indication, and this indication is evaluated by the audio decoder such as the decoder 308 of Fig. 13 as shown by the connection line between block 304 and 308, wherein the connection line carries the error concealment indication. The audio decoder then takes necessary error concealment measures without a specific audio decoding operation (although, generally, the error concealment indication is "decoded" or interpreted) such as using data from earlier frames or heuristic procedures or other related procedures.

Fig. 17 illustrates a situation where data as discussed with respect to Fig. 1, 2, 3 or 4 or as obtained by the audio encoder in Fig. 21 are used. In step 430, the highest frequency that corresponds to the correct most significant bits is determined as the concealment border. Alternatively, a maximum of the highest frequency of the MSB and the LSB/sign bits can also be determined 434 as the concealment border.

In step 432, higher frequency values are synthesized either completely or using probably obtained LSB or sign bits: Although higher frequency MSB bits are corrupted, nevertheless LSB or sign bits are still available for such spectral values corresponding to corrupted MSB bits. In step 436, the erroneous data are also synthesized when erroneous data cover LSB or sign bits of lower frequency values than the concealment border. The output of the synthesis procedure is synthesis/concealment data in the form of spectral values for the concealment spectral range schematically illustrated in Fig. 16.

Fig. 18 illustrates a preferred implementation of the frame reader 306 of Fig. 13. The frame reader preferably comprises an input buffer 442, and an output buffer 452 or a direct output to the audio decoder. The frame reader 306 comprises a controller 440, a first controlled input pointer 444 that operates in a first reading direction and a second controlled input pointer 446 that operates in a second reading direction. The first controlled input pointer 444 may feed the first writing pointer 448 and the second controlled input pointer 446 may feed the second writing pointer 450.

As the corresponding controller on the encoder-side illustrated at 800 in Fig. 8, the controller 440 of the pointer implementation on the receiver side also receives an information on the number of codewords in the second predefined subset or information on the total number or capacity of the psychoacoustically less important information units.

Additionally, the controller 440 may receive an indication of erroneous codewords of the second subset from the error protection processor in case the data from the second predefined subset of the codewords are used as much as possible as discussed with respect to Fig. 17. If this is not the case, and if it is determined that at least one codeword of the second predefined subset of the codewords is erroneous and, therefore, all spectral data that are covered by the second predefined subset of the codewords are generated by the error concealment operation as synthesis or concealment data, any control from the error protection processor to the controller 440 is not required.

Nevertheless, the procedure of the controller 440 is similar to the controller 800 of Fig. 8, but the notion of writing and reading is exchanged. Particularly, the controller 440 in Fig. 18 controls the writing points for example for a data exchange or by writing in increments. The data exchange is done in case of arrangement as is discussed in the context of Fig. 19, while the incrementation/decrementation procedure is performed for the direct decoding illustrated in Fig. 20.

The controller 440 sets the first and second pointers' start position/addresses in a frame for the first amount of information units and then synchronously increments/decrements the input (reading) pointers. The controller 440 detects that all codewords of the first subset are read from the input buffer and subsequently sets the first and second pointers' start position/addresses for the second amount of information units and synchronously increments/decrements the reading pointers until all remaining data are read.

In a first example case, the data written into the output buffer 452 are requested from the decoder, since only the decoder, and particularly the entropy decoder and/or residual decoder knows, how much information units or bits are required from which pointer. In an example, the decoder receives LSBs and Signs for a spectral line or a spectral line tuple from the first writing pointer 448 and MSBs for this same spectral line or spectral line tuple from the second writing pointer 450 as required by the entropy decoder.

In a second example case, the data are to be written into a certain frame. Then, the controller applies a syntax controlled bitstream or frame reader for reading the bits from the input buffer 442 via the pointers 444, 446 based on a certain frame syntax and the data is then written into the output buffer 452 or into a transmission device via blocks 448 and 450. In an example, the syntax controlled bitstream or frame reader reads via e.g. pointer 446 LSBs and Signs for a spectral line or a spectral line tuple and the read data is written into the buffer 452 via block 450 and the syntax controlled bitstream or frame reader reads via e.g. pointer 448 MSBs for this same spectral line or spectral line tuple and the read data is written into the output buffer 452 via block 448.

Hence, in a preferred embodiment, the audio decoder 308 of Fig. 13 typically comprises a parser and a renderer. The parser would be consist of block 71 and 72 of Fig. 22 and the renderer would include the remaining blocks of Fig. 22, since in this embodiment, one cannot determine without arithmetic or generally entropy decoding how many encoded bits are necessary for e.g. the MSB portion of a line or a line tuple or for the MSB/Sign portion of the line or the line tuple. In case of a transcoding operation, the parser is used without a subsequent renderer, since the transcoding operation outputs a frame written with a different syntax compared to the input frame.

Fig. 19 illustrates a preferred implementation, when, for example, the controller 440 or, generally, the frame reader 306 of Fig. 13 performs a rearranging operation. In step 456, the starting positions of the input (reading) pointers are determined to be so that the first predefined subset of the codewords is read. In this procedure it is, for example, the case that the second subset of the codewords is collected in adjacent codewords at the start or the end of a frame. Step 456 receives, as an input, the number of second codewords or the second amount of information units. Additionally, in case the second predefined subset of the codewords is located at the end of the frame, the total number of information units per frame is required by step 456 as well.

In step 458, the procedure stops, when the reading pointers meet each other and, at this event, the meeting location in the frame is determined. In step 460, the data up to the starting position for reading is exchanged with the data between this position and the meeting position.

At the output of block 460, one has obtained the specified/standardized intermediate frame format.

Fig. 20 illustrates the procedure of the frame reader 306 for the purpose of direct decoding. Step 470 once again receives the number of second codewords or the second amount of information units. Step 470 may require the total number of information units per frame. Then, the starting positions of the input (reading) pointers are determined to read the first predefined subset. Step 470 as well as step 456 control the reading pointers 444, 446. In step 472, the procedure stops, when the reading pointers meet each other and the meeting location is obtained. In step 474, the reading is continued over the remaining part from both sides, where the starting point for reading in the opposite direction is the determined starting point in the first step. At the output of block 474, one obtains the data for the audio decoder for the direct decoding application.

Although some aspects have been described in the context of an apparatus, it is clear that these aspects also represent a description of the corresponding method, where a block or device corresponds to a method step or a feature of a method step. Analogously, aspects described in the context of a method step also represent a description of a corresponding block or item or feature of a corresponding apparatus.

Depending on certain implementation requirements, embodiments of the invention can be implemented in hardware or in software. The implementation can be performed using a digital storage medium, for example a floppy disk, a DVD, a CD, a ROM, a PROM, an EPROM, an EEPROM or a FLASH memory, having electronically readable control signals stored thereon, which cooperate (or are capable of cooperating) with a programmable computer system such that the respective method is performed.

Some embodiments according to the invention comprise a data carrier having electronically readable control signals, which are capable of cooperating with a programmable computer system, such that one of the methods described herein is performed.

Generally, embodiments of the present invention can be implemented as a computer program product with a program code, the program code being operative for performing one of the methods when the computer program product runs on a computer. The program code may for example be stored on a machine readable carrier.

Other embodiments comprise the computer program for performing one of the methods described herein, stored on a machine readable carrier or a non-transitory storage medium.

In other words, an embodiment of the inventive method is, therefore, a computer program having a program code for performing one of the methods described herein, when the computer program runs on a computer.

A further embodiment of the inventive methods is, therefore, a data carrier (or a digital storage medium, or a computer-readable medium) comprising, recorded thereon, the computer program for performing one of the methods described herein.

A further embodiment of the inventive method is, therefore, a data stream or a sequence of signals representing the computer program for performing one of the methods described herein. The data stream or the sequence of signals may for example be configured to be transferred via a data communication connection, for example via the Internet.

A further embodiment comprises a processing means, for example a computer, or a programmable logic device, configured to or adapted to perform one of the methods described herein.

A further embodiment comprises a computer having installed thereon the computer program for performing one of the methods described herein.

In some embodiments, a programmable logic device (for example a field programmable gate array) may be used to perform some or all of the functionalities of the methods described herein. In some embodiments, a field programmable gate array may cooperate with a microprocessor in order to perform one of the methods described herein. Generally, the methods are preferably performed by any hardware apparatus.

The above described embodiments are merely illustrative for the principles of the present invention. It is understood that modifications and variations of the arrangements and the details described herein will be apparent to others skilled in the art. It is the intent, therefore, to be limited only by the scope of the impending patent claims and not by the

specific details presented by way of description and explanation of the embodiments herein.

Bibliography

[1] "ISO/IEC14496-3 MPEG-4 Information technology — Coding of audio-visual objects - Part 3: Audio," 2009.

[2] "ETSI ES 201 980 Digital Radio Mondiale; System Specification," 2014.

[3] "ETSI TR 103 590 V1.1.1 (2018-09) "Digital Enhanced Cordless Telecommunications (DECT); Study of Super Wideband Codec in DECT for narrowband, wideband and super-wideband audio communication including options of low delay audio connections (lower than 10 ms framing)".

Claims

1. Audio transmitter processor for generating an error protected frame using encoded audio data of an audio frame, the encoded audio data for the audio frame comprising a first amount of information units and a second amount of information units, the audio transmitter processor comprising:

a frame builder (506) for building a codeword frame having a codeword raster defining reference positions for a predefined total number of codewords, wherein the frame builder (506) is configured

to write the information units of the first amount of information units starting at reference positions of a first predefined subset of the codewords; and

to write the information units of the second amount of information units starting at reference positions of a second predefined subset of the codewords,

wherein the frame builder (506) is configured to determine a border between the first amount of information units and the second amount of information units so that a starting information unit of the second amount of information units coincides with a codeword border; and

an error protection coder (508)

for processing one or more of the codewords of the first predefined subset of the codewords to obtain a first processing result or for processing one or more of the codewords of the second predefined subset of the codewords to obtain a second processing result and for adding the first processing result or the second processing result to the predefined number of codewords to obtain a plurality of processed codewords representing the error protected frame, or

for processing the codewords of the first predefined subset of the codewords or of the second predefined subset of the codewords individually to obtain a plurality of processed codewords representing the error protected frame.

2. Audio transmitter processor of claim 1,

wherein the frame builder (506) is configured to determine (601) the border between the first amount of information units and the second amount of information units based on an information capacity of the second predefined subset, so that the information capacity of the second predefined subset of the codewords is equal to the amount of information units of the second amount of information units.

3. Audio transmitter processor of claim 1 or 2, further comprising:

a source encoder (500) for generating the first and the second amounts of information units using a predefined time portion of an audio signal, wherein the information units comprise a number of obligatory information units and a variable number of residual information units, wherein the source encoder (500) is configured for using a variable length coding rule (60) resulting in a signal-dependent number of the obligatory information units for the predefined time portion,

wherein the frame builder (506) is configured to build the codeword frame so that the codeword frame has a fixed size of information units, and

wherein the audio encoder is configured to determine the variable number of residual information units as an information amount being equal to the difference derived from the fixed size of information units for the codeword frame and the number of obligatory information units.

4. Audio transmitter processor of one of the preceding claims,

wherein the frame builder (506) is configured to determine the border between the first amount of information units and the second amount of information units signal-adaptively from codeword frame to codeword frame, so that, depending on the audio signal for a codeword frame, the border represents an encoded information unit relating to different audio information of the codeword frame or being interpreted differently by an audio decoder.

5. Audio transmitter processor of one of the preceding claims,

wherein the frame builder (506) is configured to use, as the second subset of the codewords, a second predefined number of adjacent codewords at one side of the codeword frame of the encoded audio data, to use, as the first predefined subset of the codewords, a first predefined number of adjacent codewords at another side of the codeword frame of the encoded audio data, and wherein a sum of the first predefined number and the second predefined number is equal to the predefined total number of codewords, or

wherein at least one codeword of the second predefined subset of the codewords is located between two codewords of the first predefined subset of the codewords or vice versa.

6. Audio transmitter processor in accordance with one of the preceding claims, wherein the frame builder (506) is configured

to write the first amount of information units using reference positions of the first predefined subset of the codewords into the first predefined subset of the codewords,

as soon as the first predefined subset of the codewords is filled (601), to write (602) the second amount of information units using reference positions of the second predefined subset of the codewords, and

wherein the frame builder (506) is configured to determine the border (601) as the information unit written into a last codeword of the first predefined subset of the codewords or as the first information unit written at a reference position of a first codeword of the second predefined subset of the codewords.

7. Audio transmitter processor of one of the preceding claims,

wherein the frame builder (506) is configured to write the information units of the first amount of information units in a first writing direction starting at reference positions of at least two codewords of the first predefined number of codewords,

and in a second opposite writing direction starting at reference positions of at least two other codewords of the first predefined number of codewords, or

wherein the frame builder (506) is configured to write the information units of the second amount of information units in a first writing direction starting at reference positions of at least two codewords of the second predefined number of codewords and in a second opposite writing direction starting at reference positions of at least two other codewords of the second predefined number of codewords.

8. Audio transmitter processor of one of the preceding claims,

wherein the frame builder (506) is configured to write information units of the first amount of information units and the second amount of information units, wherein at least a majority of the information units of the first amount of information units is psychoacoustically more important than a majority of the information units of the second amount of information units, or

wherein only a partial frame loss concealment is envisioned in an audio receiver processor when only information units in the second amount of information units are determined as corrupted and wherein a full frame loss concealment is envisioned in the audio receiver processor when information units in the first amount of information units are determined as corrupted.

9. Audio transmitter processor of one of the preceding claims, further comprising a source encoder (500) for generating at least two categories of information units, the at least two categories being selected from the group consisting of fixed length side information, variable length side information, temporal noise shaping information, one or more most significant bits of a spectrum of a first frequency portion of the spectrum, one or more most significant bits of a second frequency portion having higher frequencies than the first frequency portion, one or more least significant bits or sign bits of a first frequency portion, one or more least significant bits or sign bits of a second frequency portion, the second frequency portion having higher frequencies than the first frequency portion, and residual bits,

wherein, if generated as a category by the source encoder (500), the fixed length side information, the variable length side information, the temporal noise shaping

information, the one or more most significant bits of a spectrum of the first frequency portion, and the least significant bits or sign bits of a first frequency portion are located in the first amount of information units, and wherein, if generated as a category by the source encoder (500), at least the one of the one or more most significant bits of the second frequency portion, the one or more least significant bits or sign bits of a second frequency portion and the residual bits are in the second amount of the information units.

10. Audio transmitter processor of one of the preceding claims,

wherein the frame builder (506) is configured to use a first writing pointer (810) and a second writing pointer (812),

wherein the frame builder (506) is configured to increment the first writing pointer (810) in a direction from a lower information unit position in the codeword frame to a higher information unit position in the codeword frame,

wherein the frame builder (506) is configured to decrement the second writing pointer (812) in a direction from a higher information position in the codeword frame to a lower information position in the codeword frame,

wherein the frame builder (506) is configured to start the first writing pointer (810) at a reference position indicating a start of a first codeword in a predefined sequence of codewords of the first predefined subset of the codewords,

wherein the frame builder (506) is configured to start the second writing pointer (812) at a reference position indicating an end of a last codeword in the predefined first sequence of codewords of the predefined first set of codewords,

wherein the frame builder (506) is configured to write information units belonging to one or more first categories of information units from the first amount of information units at the positions indicated by the first writing pointer (810), and

wherein the frame builder (506) is configured to write information units belonging to one or more second categories of information units from the first amount of information units at the position indicated by the second writing pointer (812) until

the first and second writing pointers (810, 812) meet each other at a meeting information unit position,

wherein the information unit at the meeting information unit position represents the border between the first and the second amount of information units.

11. Audio transmitter processor of claim 10,

wherein the frame builder (506) is configured to set the first pointer (810) to a reference position indicating the start of a first codeword of a second predefined sequence of codewords of the second predefined subset of the codewords,

to set the second writing pointer (812) to a reference position indicating the end of a last codeword of the second predefined sequence of codewords of the second predefined subset of the codewords, and

to write information units of one or more third categories from the second amount of information units using the first writing pointer (810) and to write information units of one or more fourth categories from the second amount of information units using the second writing pointer (812).

12. Audio transmitter processor of claim 10 or 11, wherein the one or more first categories of information units comprise at least one of temporal noise shaping information, and/or one or more most significant bits of a spectrum of a first frequency portion, or

wherein the one or more second categories of information units comprise at least one of fixed or variable length side information, and least significant bits or sign bits of the first frequency portion, or

wherein the one or more third categories of information units comprise at least one of most significant bits of a second frequency portion, or

wherein the one or more fourth categories of information units comprise at least one of least significant bits or sign bits of the second frequency portion and residual data.

13. Audio transmitter processor of claim 10 to 12,

wherein the frame builder (506) is configured to use, as the one or more first category, most significant bits derived from spectral values ordered in an ascending order with respect to frequency until a spectral value or a combination of spectral values at the meeting information unit,

to use, as the one or more second category, least significant bits or sign bits derived from spectral values until the spectral value or the combination of spectral values of the meeting information unit;

to use, as the one or more third category, one or more most significant bits associated with a frequency at or above the value at the meeting information unit; and

to use, as the fourth category, least significant bits or sign bits associated with frequencies at or above the frequency value associated with the meeting information unit.

14. Audio transmitter processor of claim 13,

wherein the frame builder (506) is configured to write the one or more first categories and the one or more second categories, or to write the one or more third categories and the one or more fourth categories in synchrony with each other, so that the at least one or more most significant bits of a spectral value or a combination of spectral values are written by the first writing pointer (810) and the one or more least significant bits or sign bits of the same spectral value or of the same combination of spectral values are written, before information units from a further spectral value or a further combination of spectral values is written by the writing pointers (810, 812).

15. Audio transmitter processor of one of the preceding claims, further comprising a source encoder (500) for generating the encoded audio data in an intermediate representation extending from a start position to an end position, wherein the frame builder (506) is configured to rearrange (502) the encoded audio data in the

intermediate representation into the order represented by the first and second predefined subsets of the codewords.

16. Audio transmitter processor of claim 15,

wherein the frame builder (506) is configured to read and parse (202), from the start position and the end position, until a predefined amount of information units represented by a capacity of the first predefined subset of the codewords is obtained, and

to write (206) the information units read and parsed into the first predefined subset of the codewords and to write remaining information units between the intermediate information unit obtained when the predefined amount determined by the capacity is read and parsed, into the second predefined subset of the codewords.

17. Audio transmitter processor of claim 16, wherein encoded temporal noise shaping data, side information data and the most significant data portion and the least significant data portion are read and written to the first predefined subset of the codewords and wherein the most significant data portion and the least significant data portion of spectral values above a specific frequency and residual data are written into the second predefined subset of the codewords.

18. Audio transmitter processor of one of the preceding claims,

wherein a number of codewords of the second predefined subset of the codewords is derived from a characteristic of the error protection code, and/or a total size of the plurality of processed codewords.

19. Audio transmitter processor of one of the preceding claims, wherein the error protection coder (508) is configured

to calculate (210) the first processing result as a first Hash value,

to calculate (212) the second processing result as a second Hash value,

to add (214) the first Hash value to the first predefined subset of the codewords and to add (214) the second Hash value to the second predefined subset of the codewords, and

to apply a codeword-wise block code to obtain the error protected codewords.

20. Audio transmitter processor of one of the preceding claims,

further comprising an audio encoder (500) having an arithmetic encoder (60) generating minimum size data portions in a first granularity of information units,

wherein the error protection coder (508) is configured to operate in a second granularity of information units,

wherein the first granularity is different from the second granularity.

21. Audio transmitter processor of claim 20,

wherein the first granularity is greater than the second granularity, and

wherein at a reference position of a first codeword of the first predefined subset an adaption amount of information units corresponding to a granularity difference between the first and the second granularity is located.

22. Audio transmitter processor of claim 21, wherein the first granularity is greater than the second granularity and in an integer ratio, and wherein the adaption amount of information units together with data located at a reference position of a codeword of the second predefined subset of the codewords represent the minimum size data portion output by the arithmetic encoder (60).

23. Audio receiver processor for processing an error protected frame, comprising:

a receiver interface (300) for receiving the error protected frame to obtain a received error protected frame;

an error protection processor (302) for processing the received error protected frame to obtain an encoded audio frame, wherein the error protection processor is configured to check whether a codeword of a first predefined subset of codewords of the encoded audio frame comprises an error, and

an error concealer or an error concealment indicator (304) configured to perform a frame loss concealment operation or to generate a frame loss concealment indication in case of a detected error in the first predefined subset of the codewords.

24. Audio receiver processor of claim 23,

wherein the error protection processor (302) is configured to check (406) whether a second predefined subset of codewords of the encoded audio frame comprises an error, and

wherein the error concealer or the error concealment indicator (304) is configured to perform a partial frame loss concealment operation (410) or to generate a partial frame loss concealment indication for concealing the error in the second predefined subset of the codewords.

25. Audio receiver processor of claim 23 or 24, further comprising an audio decoder (308) for decoding the encoded audio data included in the first predefined subset of the codewords, wherein the audio decoder (308) is configured to be controlled by the error protection processor (302) to perform decoding when the error protection processor (302) does not locate an error in the first predefined subset of the codewords.

26. Audio receiver processor of one of the claims 23 to 25,

wherein the error protection processor (302) is configured to apply a block decoding operation to the received error protected frame to obtain a decoded first predefined subset of the codewords and a decoded second predefined subset of the codewords,

wherein, if the block decoding operation does not signal that an error exists, the error protection processor is configured

to calculate (402) a first Hash value from the first predefined subset of the codewords, and

to compare (404, 406) the first Hash value to a received first Hash value,

wherein the error protection processor is configured to determine that an error exists in the first predefined subset of the codewords when the first Hash value and the received first Hash value do not coincide with each other or to determine that the first predefined subset of the codewords is error-free when the first Hash value and the received first Hash value coincide.

27. Audio receiver processor of one of the claims 23 to 26,

wherein the error protection processor (302) is configured to apply a block decoding operation to the received error protected frame to obtain a decoded first predefined subset of the codewords and a decoded second predefined subset of the codewords,

wherein, if the block decoding operation does not signal that an error exists, the error protection processor is configured

to calculate (406) a second Hash value from the second predefined subset of the codewords, and

to compare (404, 406) the second Hash value to a received second Hash value,

wherein the error protection processor is configured to determine that an error exists in the second predefined subset of the codewords when the second Hash value and the received second Hash value do not coincide with each other or to determine that the second predefined subset of the codewords is error-free when the second Hash value and the received second Hash value coincide.

28. Audio receiver processor of claim 26 or 27, wherein the block decoding operation comprises a Reed-Solomon decoding operation.

29. Audio receiver processor of one of the claims 23 to 28,

wherein the error protection processor (302) is configured to check whether the second predefined subset of the codewords comprises an error only when the check of the first predefined subset of the codewords did not detect an error.

30. Audio receiver processor of one of the claims 23 to 29,

wherein the error protection processor (302) is configured to detect, whether one or more specific codewords of the second predefined subset of the codewords comprises an error, and whether any codeword of the second predefined subset of the codewords is an error-free codeword, and

wherein the error concealer (403) is configured to generate substitute data for the one or more specific codewords or an error concealment indication, and

wherein the audio receiver processor comprises an audio decoder (308) for decoding audio data included in the first predefined subset of the codewords and audio data included in one or more of the error-free codewords of the second predefined subset of the codewords, and the substitute data for the one or more specific codewords (428), or

wherein the audio receiver processor comprises an audio decoder (308) for decoding audio data included in the first predefined subset of the codewords and audio data included in one or more of the error-free codewords of the second predefined subset of the codewords, and wherein the audio decoder (308) is configured to receive the error concealment indication and to take an error concealment measure in response to the error concealment indication.

31. Audio receiver processor of one of the claims 23 to 30, further comprising a frame reader (306) for reading the encoded audio frame in accordance with a predefined frame-reading procedure identifying the first predefined subset of the codewords and a second predefined subset of the codewords.

32. Audio receiver processor of claim 31,

wherein the frame reader (306) is configured to read information units of a first amount of information units starting at reference points of codewords of the first predefined subset of the codewords until the information units from all codewords of the first predefined subset of the codeword are read, and

to read information units of a second amount of information units starting at reference points of codewords of the second predefined subset of the codewords,

further comprising an audio decoder, wherein the audio decoder (308) is configured to decode a concatenation of the first amount of information units and the second amount of information units in order to obtain decoded audio data for the encoded audio frame.

33. Audio receiver processor of claim 31 or 32,

wherein the frame reader (306) is configured to use, as the second subset of the codewords, a second predefined number of adjacent codewords at one side of the codeword frame of the encoded audio data, to use, as the first predefined subset of the codewords, a first predefined number of adjacent codewords at another side of the codeword frame of the encoded audio data, and wherein a sum of the first predefined number and the second predefined number is equal to the predefined total number of codewords, or

wherein at least one codeword of the second predefined subset of the codewords is located between two codewords of the first predefined subset of the codewords or vice versa.

34. Audio receiver processor of one of claims 31 to 33,

wherein the frame reader (306) is configured to read the information unit of the first amount of information units in a first reading direction starting at reference positions of at least two codewords of the first predefined number of codewords,

and in a second opposite reading direction starting at reference positions of at least two other codewords of the first predefined number of codewords, or

wherein the frame reader (306) is configured to read the information units of the second amount of information units in a first reading direction starting at reference positions of at least two codewords of the second predefined number of codewords and in a second opposite reading direction starting at reference positions of at least two other codewords of the second predefined number of codewords.

35. Audio receiver processor of one of claims 31 to 34,

wherein the frame reader (306) is configured to read information units of the first amount of information units and the second amount of information units, wherein at least a majority of the information units of the first amount of information units is psychoacoustically more important than a majority of the information units of the second amount of information units, or

wherein only a partial frame loss concealment is performed by the error concealer when only information units in the second amount of information units are determined as corrupted and wherein a full frame loss concealment is performed by the error concealer when information units in the first amount of information units are determined as corrupted.

36. Audio receiver processor of one of claims 31 to 35, further comprising a source decoder (308) for decoding at least two categories of information units, the at least two categories being selected from the group consisting of fixed length side information, variable length side information, temporal noise shaping information, one or more most significant bits of a spectrum of a first frequency portion of the spectrum, one or more most significant bits of a second frequency portion having higher frequencies than the first frequency portion, one or more least significant bits or sign bits of a first frequency portion, one or more least significant bits or sign bits of a second frequency portion, the second frequency portion having higher frequencies than the first frequency portion, and residual bits,

wherein, if available for decoding by the source decoder (308) as a category, the fixed length side information, the variable length side information, the temporal

noise shaping information, the one or more most significant bits of a spectrum of the first frequency portion, and the least significant bits or sign bits of a first frequency portion are located in the first amount of information units, and wherein, if available for decoding by the source decoder (308) as a category, at least one of the one or more most significant bits of the second frequency portion, the one or more least significant bits or sign bits of a second frequency portion and the residual bits are in the second amount of the information units.

37. Audio receiver processor of one of claims 31 to 36, wherein the frame reader (306) is configured

to use a first reading pointer (444) and a second reading pointer (446),

to increment the first reading pointer (444) in a direction from a lower information unit position in the codeword frame to a higher information unit position in the codeword frame,

to decrement the second reading pointer (446) in a direction from a higher information position in the codeword frame to a lower information position in the codeword frame,

to start the first reading pointer (444) at a reference position indicating a start of a first codeword in a predefined sequence of codewords of the first predefined subset of the codewords,

to start the second reading pointer (446) at a reference position indicating an end of a last codeword in the predefined first sequence of codewords of the predefined first set of codewords,

to read information units belonging to one or more first categories of information units from the first amount of information units at the positions indicated by the first reading pointer (444), and

to read information units belonging to one or more second categories of information units from the first amount of information units at the position indicated by the second reading pointer (446) until the first and second

reading pointers (444, 446) meet each other at a meeting information unit position,

wherein the information unit at the meeting information unit position represents a border between the first and the second amount of information units.

38. Audio receiver processor of claim 37,

wherein the frame reader (306) is configured to set the first reading pointer (444) to a reference position indicating a start of a first codeword of a second predefined sequence of codewords of the second predefined subset of the codewords,

to set the second reading pointer (446) to a reference position indicating the end of a last codeword of the second predefined sequence of codewords of the second predefined subset of the codewords, and

to read information units of one or more third categories from the second amount of information units using the first reading pointer (444) and to read information units of one or more fourth categories from the second amount of information units using the second reading pointer (446).

39. Audio receiver processor of claim 37 or 38,

wherein the frame reader (306) is configured to use, as the one or more first category, most significant bits derived from spectral values ordered in an ascending order with respect to frequency until a spectral value or a combination of spectral values at the meeting information unit,

to use, as the one or more second category, least significant bits or sign bits derived from spectral values until the spectral value or the combination of spectral values of the meeting information unit;

to use, as the one or more third category, one or more most significant bits associated with a frequency at or above the value at the meeting information unit; and

to use, as the fourth category, least significant bits or sign bits associated with frequencies at or above the frequency value associated with the meeting information unit.

40. Audio receiver processor of claim 39,

wherein the frame reader (306) is configured to read the one or more first categories and the one or more second categories, or to read the one or more third categories and the one or more fourth categories in synchrony with each other, so that the at least one or more most significant bits of a spectral value or a combination of spectral values are read by the first reading pointer (444) and the one or more least significant bits or sign bits of the same spectral value or of the same combination of spectral values are read, before information units from a further spectral value or a further combination of spectral values is read by the reading pointers (444, 446).

41. Audio receiver processor of one of claims 31 to 40,

wherein the frame reader (306) is configured to use a first reading pointer (444) with a first reading direction and a second pointer (446) with a second reading direction, wherein the encoded audio frame has a frame start position and a frame end position, wherein the frame reader (306) is configured to set the first pointer at the first pointer start position derived from a number of information units included in the second predefined subset of the codewords and to set the second pointer (446) at the frame end position, wherein the first predefined subset of the codewords are all located at a side of the encoded audio frame having the frame end position,

wherein the frame reader (306) is configured

to read using incrementing the first reading pointer (444) and decrementing the second reading pointer (446) in corresponding reading directions,

to stop, when the first and the second reading pointers (444, 446) meet each other at a meeting position, and

to exchange information units between the frame start position and the first pointer start position with the information units between the first pointer start position and the meeting position so that a rearranged encoded audio frame comprises, in the following order, from the start of the rearranged audio frame, information units read between the first pointer start position and the meeting point, information units of the encoded audio frame between the frame start position and the first pointer start position, and information units of the encoded audio frame between the meeting position and the frame end position, and

further comprising an audio decoder (308) preset to decode the rearranged encoded audio frame having the encoded audio data in the order of the rearranged encoded audio frame, or

further comprising a transmitter device for transmitting the rearranged encoded audio frame to a remote location.

42. Audio receiver processor of one of claims 31 to 40,

wherein the frame reader (306) is configured to use a first reading pointer (444) with a first reading direction and a second pointer (446) with a second reading direction, wherein the encoded audio frame has a frame start position and a frame end position, wherein the frame reader (306) is configured to set the second pointer (446) at a second pointer start position derived from a number of information units included in the second predefined subset of the codewords and a total number of information units of the encoded audio frame and to set the first reading pointer (444) at the frame start position, wherein the first predefined subset of the codewords are all located at a side of the encoded audio frame having the frame start position,

wherein the frame reader (306) is configured

to read using incrementing the first reading pointer (444) and decrementing the second reading pointer (446) in corresponding reading directions,

to stop, when the first and the second reading pointers (444, 446) meet each other at a meeting position, and

to exchange information units between the frame end position and the second pointer start position with the information units between the second pointer start position and the meeting position, so that a rearranged encoded audio frame comprises, in the following order from a start of the rearranged encoded audio frame, information units between the frame start position and the meeting position, information units of the encoded audio frame between the frame end position and the second pointer starting position, and information units of the encoded audio frame between the meeting position and the second pointer start position.

43. Audio receiver processor of one of claims 23 to 42, further comprising an audio decoder (308) having an arithmetic decoder (72) receiving a minimum data size portion in a first granularity of information units,

wherein the error protection processor (302) is configured to operate in a second granularity, wherein the first granularity is greater than the second granularity,

wherein, at a reference position of a first codeword of the first predefined subset of the codewords, an adaption amount of information units corresponding to a granularity difference between the first and the second granularities is located,

wherein the adaption amount of information units together with data located at a reference position of a codeword of the second predefined subset forms the minimum data size portion to be input into the arithmetic decoder (72), and

wherein the frame reader (306) is configured to start reading information units for the first predefined subset of the codewords at a position in the first codeword determined by the reference position of the first codeword plus the size of the adaption amount.

44. Method of generating an error protected frame using encoded audio data of an audio frame, the encoded audio data for the audio frame comprising a first amount

of information units and a second amount of information units, the method comprising:

building a codeword frame having a codeword raster defining reference positions for a predefined total number of codewords, wherein the building comprises:

writing the information units of the first amount of information units starting at reference positions of a first predefined subset of the codewords; and

writing the information units of the second amount of information units starting at reference positions of a second predefined subset of the codewords,

determining a border between the first amount of information units and the second amount of information units so that a starting information unit of the second amount of information units coincides with a codeword border; and

performing an error protection processing comprising:

processing one or more of the codewords of the first predefined subset of the codewords to obtain a first processing result or processing one or more of the codewords of the second predefined subset of the codewords to obtain a second processing result and adding the first processing result or the second processing result to the predefined number of codewords to obtain a plurality of processed codewords representing the error protected frame, or

processing the codewords of the first predefined subset of the codewords or of the second predefined subset of the codewords individually to obtain a plurality of processed codewords representing the error protected frame.

45. Method of processing an error protected frame, comprising:

receiving the error protected frame to obtain a received error protected frame;

performing an error protection processing with the received error protected frame to obtain an encoded audio frame, wherein the error protection processing

comprises checking whether a codeword of a first predefined subset of codewords of the encoded audio frame comprises an error, and

performing a frame loss concealment operation in case of a detected error in the first predefined subset of the codewords.

46. Computer program for performing, when running on a computer or a processor, a method of claim 44 or claim 45.