

ENTROPY CODING WITH COMPACT CODEBOOKSField of the Invention

5 The present invention relates to the encoding/decoding of information values and in particular to entropy coding using compact codebooks to generate an efficient code.

Background of the Invention and Prior Art

10

In recent times, the multi-channel audio reproduction technique is becoming more and more important. This may be due to the fact that audio compression/encoding techniques such as the well-known mp3 technique have made it possible to
15 distribute audio records via the Internet or other transmission channels having a limited bandwidth. The mp3 coding technique has become so famous because of the fact that it allows distribution of all the records in a stereo format, i.e., a digital representation of the audio record including
20 a first or left stereo channel and a second or right stereo channel.

Nevertheless, there are basic shortcomings of conventional two-channel sound systems. Therefore, the surround technique has been developed. A recommended multi-channel-surround representation includes, in addition to the two
25 stereo channels L and R, an additional center channel C and two surround channels Ls, Rs. This reference sound format is also referred to as three/two-stereo, which means three front channels and two surround channels. Generally, five transmission channels are required. In a playback environment, at least five speakers at five decent places are needed to get an optimum sweet spot in a certain distance of the five well-placed loudspeakers.

35

Several techniques are known in the art for reducing the amount of data required for transmission of a multi-channel audio signal. Such techniques are called joint stereo tech-

niques. To this end, reference is made to Fig. 9, which shows a joint stereo device 60. This device can be a device implementing e.g. intensity stereo (IS) or binaural cue coding (BCC). Such a device generally receives - as an input - at least two channels (CH1, CH2, ... CHn), and outputs at least a single carrier channel and parametric data. The parametric data are defined such that, in a decoder, an approximation of an original channel (CH1, CH2, ... CHn) can be calculated.

Normally, the carrier channel will include subband samples, spectral coefficients, time domain samples etc., which provide a comparatively fine representation of the underlying signal, while the parametric data do not include such samples of spectral coefficients but include control parameters for controlling a certain reconstruction algorithm such as weighting by multiplication, time shifting, frequency shifting, phase shifting, etc. . The parametric data, therefore, include only a comparatively coarse representation of the signal or the associated channel. Stated in numbers, the amount of data required by a carrier channel will be in the range of 60 - 70 kbit/s, while the amount of data required by parametric side information for one channel will typically be in the range of 1,5 - 2,5 kbit/s. An example for parametric data are the well-known scale factors, intensity stereo information or binaural cue parameters as will be described below.

The BCC Technique is for example described in the AES convention paper 5574, "Binaural Cue Coding applied to Stereo and Multi-Channel Audio Compression", C. Faller, F. Baumgarte, May 2002, Munich, in the IEEE WASPAA Paper "Efficient representation of spatial audio using perceptual parametrization", October 2001, Mohonk, NY, in "Binaural cue coding applied to audio compression with flexible rendering", C. Faller and F. Baumgarte, AES 113th Convention, Los Angeles, Preprint 5686, October 2002 and in "Binaural cue coding - Part II: Schemes and applications", C. Faller

and F. Baumgarte, IEEE Trans. on Speech and Audio Proc., volume level. 11, no. 6, Nov. 2003.

In BCC encoding, a number of audio input channels are converted to a spectral representation using a DFT (Discrete Fourier Transform) based transform with overlapping windows. The resulting uniform spectrum is divided into non-overlapping partitions. Each partition approximately has a bandwidth proportional to the equivalent rectangular bandwidth (ERB). The BCC parameters are then estimated between two channels for each partition. These BCC parameters are normally given for each channel with respect to a reference channel and are furthermore quantized. The transmitted parameters are finally calculated in accordance with prescribed formulas (encoded), which may also depend on the specific partitions of the signal to be processed.

A number of BCC parameters do exist. The ICLD parameter, for example, describes the difference (ratio) of the energies contained in 2 compared channels. The ICC parameter (inter-channel coherence/correlation) describes the correlation between the two channels, which can be understood as the similarity of the waveforms of the two channels. The ICTD parameter (inter-channel time difference) describes a global time shift between the 2 channels whereas the IPD parameter (inter-channel phase difference) describes the same with respect to the phases of the signals.

One should be aware that, in a frame-wise processing of an audio signal, the BCC analysis is also performed frame-wise, i.e. time-varying, and also frequency-wise. This means that, for each spectral band, the BCC parameters are individually obtained. This further means that, in case a audio filter bank decomposes the input signal into for example 32 band pass signals, a BCC analysis block obtains a set of BCC parameters for each of the 32 bands.

A related technique, also known as parametric stereo, is described in J. Breebaart, S. van de Par, A. Kohlrausch, E. Schuijers, "High-Quality Parametric Spatial Audio Coding at Low Bitrates", AES 116th Convention, Berlin, Preprint 6072, May 2004, and E. Schuijers, J. Breebaart, H. Purnhagen, J. Engdegard, "Low Complexity Parametric Stereo Coding", AES 116th Convention, Berlin, Preprint 6073, May 2004.

Summarizing, recent approaches for parametric coding of multi-channel audio signals ("Spatial Audio Coding", "Binaural Cue Coding" (BCC) etc.) represent a multi-channel audio signal by means of a downmix signal (could be monophonic or comprise several channels) and parametric side information ("spatial cues") characterizing its perceived spatial sound stage. It is desirable to keep the rate of side information as low as possible in order to minimize overhead information and leave as much of the available transmission capacity for the coding of the downmix signals.

One way to keep the bit rate of the side information low is to losslessly encode the side information of a spatial audio scheme by applying, for example, entropy coding algorithms to the side information.

Lossless coding has been extensively applied in general audio coding in order to ensure an optimally compact representation for quantized spectral coefficients and other side information. Examples for appropriate encoding schemes and methods are given within the ISO/IEC standards MPEG1 part 3, MPEG2 part 7 and MPEG4 part 3.

These standards and, for example also the IEEE paper "Noiseless Coding of Quantized Spectral Coefficients in MPEG-2 Advanced Audio Coding", S. R. Quackenbush, J. D. Johnston, IEEE WASPAA, Mohonk, NY, October 1997 describe state of the art techniques that include the following measures to losslessly encode quantized parameters:

- Multi-dimensional Huffman Coding of quantized spectral coefficients
- 5 • Using a common (multi-dimensional) Huffman Codebook for sets of coefficients
- Coding the value either as a whole or coding sign information and magnitude information separately (i.e. have only Huffman codebook entries for a given absolute value which reduces the necessary codebook size, "signed" vs. "unsigned" codebooks)
- 10 • Using alternative codebooks of different largest absolute values (LAVs), i.e. different maximum absolute values within the parameters to be encoded
- 15 • Using alternative codebooks of different statistical distribution for each LAV
- 20 • Transmitting the choice of Huffman codebook as side information to the decoder
- Using "sections" to define the range of application of each selected Huffman codebook
- 25 • Differential encoding of scalefactors over frequency and subsequent Huffman coding of the result
- 30 Another technique for the lossless encoding of coarsely quantized values into a single PCM code is proposed within the MPEG1 audio standard (called grouping within the standard and used for layer 2). This is explained in more detail within the standard ISO/IEC 11172-3:93.

35

The publication "Binaural cue coding - Part II: Schemes and applications", C. Faller and F. Baumgarte, IEEE Trans. on Speech and Audio Proc., volume level. 11, no. 6, Nov. 2003

gives some information on coding of BCC parameters. It is proposed, that quantized ICLD parameters are differentially encoded

- 5 • over frequency and the result is subsequently Huffman encoded (with a one-dimensional Huffman code)
- over time and the result is subsequently Huffman encoded (with a one-dimensional Huffman code),

10

and that finally, the more efficient variant is selected as the representation of an original audio signal.

As mentioned above, it has been proposed to optimize compression performance by applying differential coding over
15 frequency and, alternatively, over time and select the more efficient variant. The selected variant is then signaled to a decoder via some side information.

20 The prior art techniques described above are useful to reduce the amount of data that, for example, has to be transmitted during an audio- or videostream. Using the described techniques of lossless encoding based on entropy-coding schemes generally results in bit streams with a non-
25 constant bit rate.

In the AAC (Advanced Audio Codec) standard, a proposal is made to reduce both, the size of the code words and the size of the underlying codebook, by using "unsigned" code-
30 books, assuming that the probability distribution of the information values to be encoded only depends on the magnitudes of the values to be encoded rather than their signs. The sign bits are then transmitted separately and can be considered as a postfix code, mapping back the coded magnitude information into the actual value (sign x magnitude).
35 Assuming for example a four-dimensional Huffman codebook, this results in saving a factor of $2^4 = 16$ (assuming that all values carry signs) in the size of the codebook.

Quite some efforts have already been made to reduce code size by entropy coding. Nonetheless, one still fights some major disadvantages using techniques of prior art. For example, when using multi-dimensional Huffman codebooks, one can achieve a decrease in the bit rate needed to transmit some encoded information. This is achieved at the cost of an increase in the size of the Huffman codebook that has to be used, since for each additional dimension, the Huffman codebook size increases by a factor of two. This is especially disadvantageous in applications where the Huffman codebook is transmitted together with the encoded information, as it is for example the case with some computer compression programs. Even if the Huffman codebook does not have to be transmitted with the data, it has to be stored in the encoder and in the decoder, needing expensive storage space, which is available only in limited quantities, especially in mobile applications for video or audio streaming or playback.

20

Summary of the invention

It is the object of the present invention to provide a concept for generation and use of a more efficient code to compress information values and to reduce the size of an underlying codebook.

According to a first aspect of the present invention, this object is achieved by an encoder for encoding information values, comprising: a grouper for grouping two or more information values into a tuple in a tuple order; a code information generator for generating order information indicating the tuple order and a code word for the tuple using an encoding rule, where the encoding rule is such that the same code word is assigned to tuples having identical information values in different orders; and an output interface for outputting the code word and in association therewith the order information.

According to a second aspect of the present invention, this object is achieved by a decoder for decoding a code word based on information values, comprising: an input interface
5 providing the code word and in association therewith an order information indicating a tuple order, being an order of two or more information values within a tuple of information values; a code processor for deriving a tuple using a decoding rule depending on an encoding rule used to create
10 the code word, where the decoding rule is such that different tuples are derived from the same code word, the different tuples having the same information values in different tuple orders, as indicated by differing order information; and a degrouper for degrouping the tuples into two or more
15 information values.

According to a third aspect of the present invention, this object is achieved by a method for encoding information values, the method comprising: grouping two or more information values into a tuple in a tuple order; generating order
20 information indicating the tuple order; generating a code word for the tuple using an encoding rule, where the encoding rule is such that the same code word is assigned to tuples having identical information values in different
25 orders; and outputting the code word and in association therewith the order information.

According to a fourth aspect of the present invention, this object is achieved by a computer program implementing the
30 above method, when running on a computer.

According to a fifth aspect of the present invention, this object is achieved by a method for decoding code words based on information values, the method comprising: providing
35 the code word and in association therewith an order information indicating a tuple order, being an order of two or more information values within a tuple of information values; deriving a tuple using a decoding rule depending on

an encoding rule used to create the code word, where the decoding rule is such that different tuples are derived from the same code word, the different tuples having the same information values in different tuple orders, as indicated by differing order information; and degrouping the tuples into two or more information values.

According to a sixth aspect of the present invention, this object is achieved by a computer program implementing the above method, when running on a computer.

According to a seventh aspect of the present invention, this object is achieved by encoded data based on information values, the encoded data comprising a code word for a tuple of two or more information values arranged in the tuple in a tuple order and an order information indicating the tuple order.

The present invention is based on the finding that an efficient code for encoding information values can be derived, when two or more information values are grouped in a tuple in a tuple order and when an encoding rule is used, that assigns the same code word to tuples having identical information values in different orders and when an order information, indicating the tuple order, is derived and associated to the code word.

For entropy coding using Huffman codes, the inventive concept described above can be implemented more efficiently. The size of a Huffman codebook depends as well on the number of possible values to be coded as on the number of jointly coded values (dimension of the Huffman codebook). In order to reduce the required storage space needed to represent a Huffman codebook in a specific application, it is advantageous to exploit symmetries in the probability distribution of the data to be coded such that one Huffman codeword represents a whole set of groups of jointly coded values with equal probability. The actual group of jointly

coded values is specified by a particular postfix code then.

5 Since the order of two or more values in a tuple is not dependent on the content represented by the values, when the values are to a certain degree uncorrelated, equal probabilities for different orders of the same values can be assumed (since the values are uncorrelated). Particularly and preferably for a variable length code, i.e. a code having
10 code words with different lengths, these equal probabilities will result in a smaller codebook and an efficient code, when the tuples with different orders of the same values are assigned to a single code word.

15 Therefore, in a preferred embodiment of the present invention, the information values that are to be encoded by a two-dimensional Huffman encoder are differentially encoded first, resulting in a differential representation having certain symmetries as explained later in more detail. After
20 that, a number of symmetry operations are applied to the differential representation, reducing the number of possible tuples to be encoded, thus also reducing the size of the required codebook.

25 Differentially encoding a distribution of values, occurring with a given probability distribution will result in a distribution of difference values that is centered around zero, having a probability distribution that is symmetric, i.e. values of the same absolute value but with different
30 signs will occur with the same probability.

The basic principle of entropy coding, as for example Huffman coding, is, that the used codebook represents a given probability distribution of information values as good as
35 possible in view of assigning the shortest possible code words to the information values occurring at most. Multi-dimensional Huffman coding follows the same principle but combines two or more information values into a tuple,

wherein the whole tuple is then associated with a single codeword.

Therefore, combining differential encoding with two-
5 dimensional Huffman encoding yields two types of symmetries, that can be made use of.

The first symmetry derives from the observation that the probability of occurrence of the tuple (a, b) is approxi-
10 mately the same as for the tuple (-a, -b). This corresponds to a point symmetry relative to the origin of a coordinate system, where the first value of a tuple defines the X-axis and the second value of a tuple defines the Y-axis.

15 The second symmetry is based on the assumption, that changing the order in which the two values in a tuple occur does not change the tuple's probability of occurrence, i.e. that (a, b) and (b, a) are equally probable. This corresponds to an axis symmetry relative to the bisectors of the coordi-
20 nate systems first and third quadrant, when the coordinate system is defined as explained above.

The two symmetries can be exploited such, that the size of a Huffman codebook is reduced by a factor of 4, meaning
25 that symmetrical tuples are assigned the same codeword. The symmetry information, i.e. the order of the original tuple and the sign of the original tuple, are indicated by an order information and a sign information and transmitted together with the codeword to allow for decoding and a reconstruction of the original tuple including the sign and the
30 order information.

Recalling the above representation of the tuples within a two-dimensional coordinate system, both symmetries together
35 can be understood as a 2-dimensional probability distribution with level curves (curves of equal probabilities) resembling ovals with the principle axis rotated by 45 degrees relative to the Cartesian coordinate system.

Making use of the two symmetries, only approximately one forth (1 quadrant) of the possible entries within the coordinate system has to be covered by Huffman codes. A two-bit postfix code determines a unique mapping between every pair of values in one of four quadrants and its corresponding pairs of values in the remaining three quadrants. Note that for pairs of values situated on either quadrant borderline, the postfix code consists of one bit only or can even be omitted in case of the pair of values situated on both border lines, i.e. in the center of the distribution.

The inventive concept will reduce the size of the Huffman codebook for data that is not showing the symmetries described above. If such data is encoded, on the one hand the size of the Huffman codebook will be small, but on the other hand the encoded representation might not be ideally compressed, since values occurring with different probabilities are represented by the same codeword, leading to a waste of bit rate since the Huffman codebook cannot be tuned to fit the data in an optimal way.

Therefore, it is preferred that the data is differentially encoded before applying symmetry treatments, since the differential encoding automatically yields advantageous symmetries. Thus, the inventive concept can be used assuring a compact representation and a small Huffman codebook for every underlying set of information values, since the disadvantage of doubling the number of possible information values by differentially encoding the information values can be balanced by using the symmetries.

Brief description of the drawings

Preferred embodiments of the present invention are subsequently described by referring to the enclosed drawings, wherein:

- Fig. 1 shows a block diagram of an inventive encoder;
- Fig. 2 shows a preferred embodiment of an inventive encoder;
- 5 Fig. 3 shows a further preferred embodiment of an inventive encoder;
- 10 Fig. 4a shows a first symmetry operation on data to be encoded;
- Fig. 4b shows a second symmetry operation on data to be encoded;
- 15 Fig. 5 shows the derivation of a symmetric representation of data;
- Fig. 6 shows a block diagram of an inventive decoder;
- 20 Fig. 7 shows a preferred embodiment of an inventive decoder;
- Fig. 8 shows a further preferred embodiment of an inventive decoder; and
- 25 Fig. 9 shows a prior art multi-channel encoder.

Detailed description of preferred embodiments

- 30 Fig. 1 shows an inventive encoder 100, the encoder 100 comprising a grouper 102 and a code information generator 104, that includes an output interface.

The information values 106 are grouped in tuples of information values 108a to 108c by the grouper. In the example shown in Fig. 1, the inventive concept is described by building tuples consisting of two information values each, i.e. by using a two-dimensional Huffman code.

35

The tuples 108a to 108c are transferred to the code information generator 104, wherein the code information generator implements an encoding rule that assigns the same code-
word to tuples having identical information values in different orders. Therefore, the tuples 108a and 108c are encoded into the same code words 110a and 110b, whereas the
tuple 108b is encoded into a different codeword 112. According to the inventive concept, differing order information 114a and 114b is generated to preserve the information of the order in which the information values are grouped inside the tuples 108a and 108c. A combination of the order information and the codeword can therefore be used to reconstruct the original tuples 108a and 108c, hence the order information is delivered in association with the codeword by the output interface. Generally, one may agree on different ordering schemes resulting in differing order information bits. In the example shown in Fig. 1, the tuples are not reordered when the values within the tuples occur in ascending order, as it is the case for the tuples 108a and 108b. If one further agrees on assigning a order information of 0 to tuples that have not been reordered, one results in the order information values as they have been assigned to the codewords in Fig. 1.

25

Fig. 2 shows a preferred embodiment of the present invention, in which the code information generator 104 comprises an order encoder 120 and an entropy encoder 122 and wherein the encoder 100 further comprises an output interface 124.

30

The order encoder 120 generates the order information 114a to 114c of the tuples (indicating the tuple order) and transfers the order information to the output interface 124. At the same time, the order encoder 120 reorders the information values within the tuples 108a to 108c to derive the tuples 126a to 126c changing the tuple order to a predefined tuple order, the predefined tuple order defin-

35

ing an encoding order of information values for groups of tuples having identical information values.

5 The reordering can for example be done, also for tuples having more than two information values, by multiple subsequent steps of exchanging the position of two information values within the tuple. After each step, it is checked, whether there exists an entry in the codebook of the order encoder for the given tuple order. If this is the case, the reordering can be stopped and the code word can be generated. If not, the above procedure is repeated until the codeword is found. The order information can then, for example, be derived from the number of exchanges necessary to derive the codeword. Similarly, the correct tuple order could be rearranged using the order information on a decoder side.

20 The entropy coder encodes the tuples 126a to 126c to derive the code words 110a, 110b, and 112 and transfers these code words to the output interface 124.

25 The output interface finally outputs the code words 110a, 110b, and 112 and in association therewith the order information 114a to 114c.

30 Fig. 3 shows another preferred embodiment of the present invention, wherein the code information generator further comprises a sign encoder 130 for deriving a sign information indicating a sign combination of the information values within the tuple.

35 The inventive encoder shown in Fig. 3 makes also use of the second symmetry, assigning the same code words to tuples having information values of same absolute value and in the same order regarding their absolute values. Therefore, the sign encoder 130 derives a sign information 132a to 132c for each of the tuples 134a to 134c, indicating the sign of the values within the tuples. The sign encoder 130 simulta-

neously changes the signs of the information values within the tuples to derive a predefined sign combination, defining an encoding sign combination for each order of absolute values within the tuple, i.e. for tuples differing only in
5 the signs of the information values.

The sign information 132a to 132c is additionally transferred to the output interface 124 that also receives order information from the order encoder 120 and the code words
10 from the entropy encoder 122. The output interface 124 then supplies the code words and in association therewith the order information and the sign information.

Fig. 4a shows, how the sign information can be derived by
15 the sign encoder 130 and how the symmetry regarding the signs can be used to reduce the size of a required Huffman codebook.

Shown are the possible values of a tuple (a, b), graphically represented by a matrix, wherein the values for a are shown on the X-axis and the possible values for b as shown on the Y-axis. The values for a and b are symmetrized around zero (for example by a previous differential encoding) and are ranging from -4 to 4 each.
25

The matrix is showing all the 81 possible combinations of the parameter values a and b. Indicated is also a first axis 150, indicating the entries of the matrix, where the sum of a and b ($a + b$) equals zero. The figure further
30 shows a second axis 152, where the difference of a and b ($a - b$) equals zero. As can be seen, the two axes 150 and 152 divide the matrix into four quadrants labeled with numbers 1 to 4.

35 The first symmetry, assuming that a combination (a, b) and a combination (-a, -b) are equally probable, is equal to a point symmetry to the origin, which is explicitly shown for two arbitrary entries 154a and 154b of the matrix .

An inventive encoder can reduce the size of a required Huffman codebook by a factor of approximately 2 by performing a symmetry operation, mirroring the entries of the third quadrant to the first quadrant and the entries of the fourth quadrant to the second quadrant. Such, the codebook entries of the third and forth quadrant can be saved, when the sign encoder 130 additionally indicates by the sign information from which quadrant the tuple was coming.

Fig. 4b shows, how the second symmetry can be made use of by the order coder 120 to reduce the size of the Huffman codebook by another factor of approximately 2. The second symmetry, meaning that tuples (a, b) and (b, a) are equally probable, is equal to a mirroring of the remaining matrix entries at the axis 152. This is for example shown for the two matrix entries 156a and 156b, showing the mirroring of the tuple $(0, 2)$ to the tuple $(2, 0)$. Therefore, by assigning corresponding entries of the second and of the first quadrant the same code words, one can save another factor of 2 in the codebook size. The order coder 120 derives the order information to preserve the information, whether the original tuple was from the first or from the second quadrant.

As illustrated by Figs. 4a and 4b, an inventive encoder incorporating a sign encoder and an order encoder allows that only one out of four quadrants has to be covered by Huffman codes. A two-bit postfix code determines a unique mapping between every pair of values in one of the four quadrants and its corresponding pair of values in the remaining three quadrants. Note that for pairs of values situated on either quadrant borderline, the postfix code consists of one bit only or can even be omitted in case of the pair of values situated on both borderlines, i.e. at the center of the matrix.

Following the symmetry operations shown in Figs. 4a and 4b, it can be fortunate to reorder (map) the pairs of values in the remaining quadrant into a quadratic matrix for practical reasons, since running an algorithm on a quadratic representation is much more convenient.

A mapping strategy incorporated by one embodiment of the present invention is shown in Fig. 5. The original matrix with the first quadrant to be mapped to a quadratic matrix 160 is shown to the left. The quadratic matrix 160 also has parameter values for parameter a on the X-axis and parameter values for parameter b on the Y-axis. The reordering following the mapping algorithm is indicated in the drawing, where each of the matrix elements is marked with a unique number. It is to be noted, that tuples whose information values sum to an even number ($a+b$ is even, which is true for all non hatched matrix elements of the first quadrant) are reordered to the lower left side of the quadratic matrix 160, whereas tuples having odd sums of information values (hatched elements) are reordered to the upper right side of the matrix 160, being hatched in Figure 5.

After the above mapping process, the tuples to be Huffman encoded are given in a quadratic representation and thus are easy to handle.

Fig. 6 shows a block diagram of an inventive decoder for decoding code words that are based on information values.

The decoder 200 comprises an input interface 202, a code processor 204, and a degrouper 206. The input interface 202 provides code words 210a, 210b and 212 and associated therewith order information 214a, 214b and 216, both are transferred to the code processor 204.

The code processor derives tuples of information values 218a, 218b, and 218c using a decoding rule, where the decoding rule is such, that different tuples are derived

from the same code word, the different tuples having the same information values in different tuple orders, as indicated by a differing order information.

- 5 Therefore, the differing tuples 218a, and 218c are derived from the same code words 210a and 210b, since having associated different order information 214a and 214b. The tuples of information values 218a to 218c are then degrouped by the degrouper 206 to yield the information values 220.

10

Fig. 7 shows a preferred inventive decoder, wherein the code processor 204 includes an entropy decoder 222 and an order decoder 224.

- 15 The entropy decoder 222 assigns, using a Huffman codebook, each of the code words 210a, 210b, and 212 to one tuple 226a to 226c. The tuples 226a to 226c are transferred to the order decoder 224 that also receives the associating order information. The order decoder 224 derives the information values modifying the tuples 226a to 226c, as indicated by the order information. The derived final tuples 228a to 228c are then transferred to the degrouper 206, that degroups the tuples 228a to 228c to derive the information values 220.

25

Fig. 8 shows another preferred embodiment of an inventive decoder 200, where the code processor 204 further includes a sign decoder 230.

- 30 The input interface provides three identical code words 232a to 232c, sign information bits 234a to 234c and order information bits 236a to 236c.

The entropy decoder 222 decodes the three identical code words 232a to 232c into three identical tuples 238a to 238c, which are then transferred to the sign decoder 230.

35

The sign decoder 230 additionally receives the sign information 234a to 234c and modifies the tuples 238a to 238c, as indicated by the sign information 234a to 234c, to derive the tuples 240a to 240c, that are now having information values with correct signs.

The tuples 240a to 240c are transferred to the order decoder 224, that additionally receives the order information 236a to 236c, and that alters the order of information values within the tuples 240a to 240c to receive the correctly ordered tuples 242a to 242c. The tuples 242a to 242c are then transferred to the degrouper 206, that derives the information values 220 by degrouping the tuples 242a to 242c.

Although the preferred embodiments of inventive encoders shown in the figures 2 and 3 propose, that the order encoder 120 and the sign encoder 130 retrieve the order or sign information and do simultaneously alter the tuples, it is alternatively possible that the sign encoder and the order encoder retrieve the order and sign information without altering the tuples. A Huffman Codebook has to be designed such then, that tuples with different sign and order informations are assigned the same code words.

Although the preferred embodiments shown in the figures detail the concept of the present invention by using two-dimensional Huffman codebooks, the inventive idea of using grouped values and making use of their symmetries can also be combined with other lossless encoding methods, especially with Huffman codebooks of higher dimensions, i.e. building tuples having more information values than two.

Figs. 4a and 4b show a way to derive the symmetry information used to allow for a smaller Huffman codebook. In a first step the third and forth quadrant of a given matrix is mapped to the first and second quadrant. It is alternatively also possible to map other combinations of two

neighboring quadrants to the remaining two quadrants, as long as the symmetry information is preserved such that the mapping can be reversed. The same is also true for the second symmetry operation shown in Fig. 4b.

5

Fig. 5 shows a way of mapping one quadrant of a matrix into a quadratic matrix allowing to conveniently use describing a codebook of reduced size. This mapping is an example for multiple possible mapping schemes, wherein the mappings only have to be such that they are unique and can be reversed.

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

While the foregoing has been particularly shown and described with reference to particular embodiments thereof, it will be understood by those skilled in the art that various other changes in the form and details may be made without departing from the spirit and scope thereof. It is to be understood that various changes may be made in adapting to different embodiments without departing from the broader concepts disclosed herein and comprehended by the claims that follow.

CLAIMS

1. Encoder for encoding information values, comprising:
 - 5 a grouper for grouping two or more information values into a tuple in a tuple order;
 - a code information generator for generating order information indicating the tuple order and a code word
 - 10 for the tuple using an encoding rule, where the encoding rule is such that the same code word is assigned to tuples having identical information values in different orders; and
 - 15 an output interface for outputting the code word and in association therewith the order information.
2. Encoder in accordance with claim 1, in which the code information generator includes an order encoder for deriving the order information and an entropy encoder for
- 20 deriving the code word using a codebook that assigns tuples having identical information values in different orders to the same code word.
- 25 3. Encoder in accordance with claim 2, in which the order encoder is operative to reorder the information values within the tuple to change the tuple order to a predefined tuple order for the tuple, the predefined tuple order defining an encoding order of information values
- 30 for groups of tuples having identical information values and in which the order encoder is operative to detect and encode a deviation of the tuple order and the encoding order; and
- 35 in which the entropy encoder is having a reduced codebook assigning a code word to tuples having information values ordered in the encoding order.

4. Encoder in accordance with claim 3, in which the order encoder is operative to reorder the information values within the tuple by exchanging a first information value with a second information value.

5

5. Encoder in accordance with claim 1, in which

the code information generator is having a sign encoder to derive a sign information indicating a sign combination of the information values within the tuple;

10

the encoding rule is such, that the same code word is assigned to tuples differing only by the sign of each information value in the tuple; and

15

the output interface is operative to output the sign information in association with the code word.

6. Encoder in accordance with claim 5, in which

20

the sign encoder is operative to change the signs of the information values within the tuple to change a sign combination to a predefined sign combination, the predefined sign combination defining an encoding sign combination for an order of information values within the tuple differing only in the individual signs of the information values; and

25

the code information generator is having an entropy encoder, the entropy encoder having a reduced codebook, assigning the same code word to each tuple having the same order of absolute values of information values.

30

7. Encoder in accordance with claim 6, in which the sign encoder is operative to change the sign of a first and of a second information value within a tuple.

35

8. Encoder in accordance with claim 1, further comprising a differential encoder for differential encoding of input values to obtain a differentially encoded representation of the input values as the information values.
- 5 9. Encoder in accordance with claim 8, in which the differentially encoded representation is a representation of the input values differentially encoded in time or in frequency.
- 10 10. Encoder in accordance with claim 1, in which the information values comprise information values describing a frame of a video signal or an audio signal.
- 15 11. Encoder in accordance with claim 1, in which the information values comprise BCC parameters describing a spatial correlation between a first and a second audio channel and in which the BCC Parameters are chosen from the following list of BCC parameters:
- 20 inter-channel coherence (ICC),
inter-channel level difference (ICLD),
inter-channel time difference (ICTD),
inter-channel phase difference (IPD).
- 25 12. Encoder in accordance with claim 1, in which the encoding rule is such, that an encoding of information values results in a sequence of code words having different lengths.
- 30 13. Decoder for decoding a code word based on information values, comprising:
- 35 an input interface for providing the code word and in association therewith an order information indicating a tuple order, being an order of two or more information values within a tuple of information values; and

5 a code processor for deriving a tuple using a decoding rule depending on an encoding rule used to create the code word, where the decoding rule is such, that different tuples are derived from the same code word associated by different order information, the different tuples having the same information values in different tuple orders.

10 14. Decoder in accordance with claim 13, in which the code processor includes

15 an entropy decoder for deriving a preliminary tuple using a codebook assigning each code-word to a preliminary tuple; and

an order decoder for deriving the tuple by reordering the information values within the preliminary tuple as indicated by the order information.

20 15. Decoder in accordance with claim 14, in which the order decoder is operative to reorder the information values of the preliminary tuple by exchanging a first information value with a second information value.

25 16. Decoder in accordance with claim 14, in which

the input interface further provides a sign information, indicating a sign combination for the information values within a tuple; and

30 in which the code processor is further comprising a sign decoder to derive the tuple from the preliminary tuple by changing the signs of the information values within the preliminary tuple as indicated by the sign information.
35

17. Decoder in accordance with claim 16, in which the sign decoder is operative to change the signs of each information value within the preliminary tuple.
- 5 18. Decoder in accordance with claim 16, in which the sign decoder is operative to change the sign of a first and of a second information value within the preliminary tuple.
- 10 19. Decoder in accordance with claim 13, further comprising a differential decoder for differential decoding of the information values to obtain a differentially decoded representation of the information values.
- 15 20. Decoder in accordance with claim 19, in which the differentially decoded representation of the information values is differentially decoded in time or in frequency.
- 20 21. Decoder in accordance with claim 13, in which the information values comprise information values describing a frame of a video signal or an audio signal.
- 25 22. Decoder in accordance with claim 13, in which the information values comprise BCC parameters describing a spatial correlation between a first and a second audio channel and in which the BCC Parameters are chosen from the following list of BCC parameters:
- 30 inter-channel coherence (ICC),
inter-channel level difference (ICLD),
inter-channel time difference (ICTD),
inter-channel phase difference (IPD).
- 35 23. A method for encoding information values, the method comprising:

grouping two or more information values into a tuple in a tuple order;

generating order information indicating the tuple order;

generating a code word for the tuple using an encoding rule, where the encoding rule is such that the same code word is assigned to tuples having identical information values in different orders; and

outputting the code word and in association therewith the order information.

24. A method for decoding code words based on information values, the method comprising:

providing the code word and in association therewith an order information indicating a tuple order, being an order of two or more information values within a tuple of information values;

deriving a tuple using a decoding rule depending on an encoding rule used to create the code word, where the decoding rule is such, that different tuples are derived from the same code word associated by different order information, the different tuples having the same information values in different tuple orders; and

degroupping the tuples into two or more information values.

25. Computer program having a program code for performing, when running on a computer, a method for encoding information values, the method comprising:

grouping two or more information values into a tuple in a tuple order;

generating order information indicating the tuple order;

5 generating a code word for the tuple using an encoding rule, where the encoding rule is such that the same code word is assigned to tuples having identical information values in different orders; and

10 outputting the code word and in association therewith the order information.

26. Computer program having a program code for performing, when running on a computer, a method for decoding code words based on information values, the method comprising:
15

providing the code word and in association therewith an order information indicating a tuple order, being an order of two or more information values within a tuple of information values;
20

deriving a tuple using a decoding rule depending on an encoding rule used to create the code word, where the decoding rule is such, that different tuples are derived from the same code word associated by different order information, the different tuples having the same information values in different tuple orders; and
25

30 degroupping the tuples into two or more information values.

27. Encoded data based on information values, the encoded data comprising a code word for a tuple of two or more information values arranged in the tuple in a tuple order and an order information indicating the tuple order.
35

28. Encoded data in accordance with claim 27, which is stored on a computer readable media.