

Encoder, Decoder, System and Methods for Encoding and Decoding

Description

5

Embodiments relate to an encoder, a decoder, a system comprising an encoder and a decoder, a method for encoding and a method for decoding. Some embodiments relate to apparatuses and methods for optimal residual quantization in source coding. Some
10 embodiments relate to a source coding scheme using entropy coding to code a quantized signal on a determined number of bits.

Entropy coding is an efficient tool for exploiting the redundancy of symbols to transmit. It is usually used in transform-based coding after the quantization of the spectral lines. By
15 exploiting an a priori probability distribution, the quantized values can be losslessly coded with a reduced number of bits. The principle lies in generating codewords for which the length is function of the symbol probability.

The bit consumption is usually only known after writing the entropy coded symbols into the
20 bit stream. It is usually problematic when optimizing the quantization stage, which needs to know the bit consumption for optimizing the rate distortion function. It is even more problematic when the bit stream has to have a constant size per frame, also known as constant bitrate, which is a requirement for most of the communication network protocols.

25 In a transform encoder, a set of scale factors defines usually the quantization, by shaping the quantization noise in frequency domain. The noise shaping is function of both the perceived distortion, usually given by a psychoacoustic model, and the engendered bit consumption. However the last factor is usually only known after fixing the quantization noise shaping. An optimization loop can be used for making the optimization converged. Nevertheless such an
30 optimization is relatively complex and the number of iterations has to be strongly limited in real applications. Moreover for reducing even more the complexity, the bit consumption is usually not fully computed but only estimated. If the final bit consumption is underestimated, the bit-stream will have to be truncated, which is avoided most of the time. Indeed an underestimation will lead to a hard truncation of the bit stream, which is equivalent to make
35 the quantization saturate. Thus, the quantization optimization is usually designed to over-estimate the bit consumption. As a consequence few bits are often unexploited in the final bit stream.

To overcome this problem, a residual (or second) quantization stage can be added after the first quantization stage for exploiting eventual unused bits. These remaining bits then can be used to refine the quantization noise. This principle is explained in the following.

5

Fig. 10 shows a block diagram of a transform encoder 10. The transform encoder 10 comprises a first quantization stage 12, a residual quantization stage 14, an entropy encoder 16, an entropy coding bit estimation unit 18, a multiplexer 20 and a transformation unit 22.

10 The transformation unit 22 is configured to transform an input signal from a time domain into a frequency domain. The first quantization stage 12 is configured to quantize the input signal in the frequency domain into a plurality of quantized spectral values q . The plurality of quantized spectral values q , the input signal in the frequency domain x and a number of remaining bits are input to the residual (or second) quantization stage 14 that is configured to
15 refine the output of the first quantization stage 12 and to provide a plurality of quantized residual values q_r . The entropy encoder 16 is configured to entropy encode the plurality of quantized spectral values q in order to obtain a plurality of entropy encoded values e . The multiplexer 20 is configured to multiplex the plurality of entropy encoded values e , the scale factors in dependence on an information provided by the first quantization stage 14 and the
20 plurality of quantized residual values delivered by the second quantization 16 in order to obtain a bit stream.

The transform encoder 10 shown in Fig. 10 is designed for delivering a certain target number of bits per frame. The quantization will be adjusted for reaching this target, but for complexity
25 reasons only an estimation of the entropy encoder bit consumption is done when adjusting the quantization steps. Moreover even if the bit estimation is very accurate it may be impossible to find a set of scale factors which lead to the expected target bits. After the first quantization stage 12, quantized values q are entropy coded. The remaining unexploited bits are then allocated to the residual quantization which will refine the output of the first
30 quantization stage 12. The residual quantization stage 14 takes as input the quantized spectral values q , the original spectral values x and a number of remaining bits. The number of remaining bits can be an estimate or the true number of remaining bits. The estimate is usually used when a local synthesis is required at the encoder side for taking for example a switching decision in a closed-loop decision fashion as it is done in AMR-WB+ (Adaptive
35 Multi-Rate Wide Band Extended). In that case, the residual coding has to be called before the eventual call of the entropy encoder 16.

In a common transform encoder 10, the residual quantization stage 14 performs a simple uniform scalar quantization of the difference of an inverse quantized input signal obtained by inverse quantizing the quantized spectral values and the original input signal. However, through rate-distortion performance analysis, it is known that the uniform quantization is only
5 optimal for memoryless and uniformly distributed sources.

Therefore, it is the object of the present invention to provide an improved residual quantization for non-memoryless and non-uniformly distributed sources.

10 This object is solved by the independent claims.

Embodiments of the present invention provide an encoder comprising a quantization stage, an entropy encoder, a residual quantization stage and a coded signal former. The quantization stage is configured to quantize an input signal using a dead zone in order to
15 obtain a plurality of quantized values. The entropy encoder is configured to encode the plurality of quantized values using an entropy encoding scheme in order to obtain a plurality of entropy encoded values. The residual quantization stage is configured to quantize a residual signal caused by the quantization stage, wherein the residual quantization stage is configured to determine at least one quantized residual value in dependence on the dead
20 zone of the quantization stage. The coded signal former is configured to form a coded signal from the plurality of entropy encoded values and the at least one quantized residual value.

Further, embodiments of the present invention provide a decoder comprising a coded signal parser, an entropy decoder and an inverse quantization stage. The coded signal parser is
25 configured to parse a coded signal in order to obtain a plurality of entropy encoded values and at least one quantized residual value. The entropy decoder is configured to decode the plurality of entropy encoded values using an entropy decoding scheme in order to obtain a plurality of quantized values. The inverse quantization stage is configured to inverse quantize the plurality of quantized values in order to obtain an output signal. Further, the inverse
30 quantization stage is configured to refine an inverse quantization level used for obtaining the output signal in dependence on the quantized residual value and a dead zone.

According to the concept of the present invention, an error between the (original) input signal and an inverse quantized signal obtained by inverse quantizing the plurality of quantized
35 values can be reduced or even optimized by a residual quantization stage at the encoder side that takes into account the dead zone which was used for quantizing the input signal and an inverse quantization stage at the decoder side that also takes into account this dead

zone when refining the inverse quantization level used for obtaining the inverse quantized signal (referred to as output signal).

Furthermore, embodiments of the present invention provide a method for encoding. The method comprises quantizing an input signal in order to obtain a plurality of quantized values using a dead zone; encoding the plurality of quantized values using an entropy encoding scheme in order to obtain a plurality of entropy encoded values; quantizing a residual signal caused by a quantization by the quantization stage and determining a plurality of quantized residual values in dependence on the dead zone of the quantization stage; and forming a bit stream from the plurality of entropy encoded values and the plurality of quantized residual values.

Moreover, embodiments of the present invention provide a method for decoding, the method comprises parsing a coded signal in order to obtain a plurality of entropy encoded values and a quantized residual value; decoding the plurality of entropy encoded values using an entropy decoding scheme in order to obtain a plurality of quantized values; inverse quantizing the plurality of quantized values using a dead zone in order to obtain an output signal; and refining an inverse quantization level used for obtaining the output signal in dependence on a dead zone and the quantized residual value.

Embodiments of the present invention are described herein making reference to the appended drawings.

Fig. 1 shows a block diagram of an encoder according to an embodiment of the present invention.

Fig. 2 shows a block diagram of a decoder according to an embodiment of the present invention.

Fig. 3 shows a block diagram of a system according to an embodiment of the present invention.

Fig. 4 shows a block diagram of the residual quantization stage according to an embodiment of the present invention.

Fig. 5 shows in a diagram inverse quantization levels and quantization thresholds used in a dead zone uniform threshold scalar quantization scheme.

Fig. 6 shows in a diagram two refined inverse quantization levels for a non-zero quantized value.

5 Fig. 7 shows in a diagram three refined inverse quantization levels for a zero quantized value.

Fig. 8 shows a flowchart of a method for encoding according to an embodiment of the present invention.

10

Fig. 9 shows a flowchart of a method for decoding according to an embodiment of the present invention.

15

Fig. 10 shows a block diagram of a conventional transform encoder using a residual quantization.

Equal or equivalent elements or elements with equal or equivalent functionality are denoted in the following description by equal or equivalent reference numerals.

20 In the following description, a plurality of details are set forth to provide a more thorough explanation of embodiments of the present invention. However, it will be apparent to those skilled in the art that embodiments of the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form rather than in detail in order to avoid obscuring embodiments of the present
25 invention. In addition, features of the different embodiments described hereinafter may be combined with each other, unless specifically noted otherwise.

Since entropy coding delivers variable length code-words, it is difficult to predict the exact bit consumption before writing the bit-stream. However the bit consumption is needed for
30 optimizing the quantization. Most of the time and for complexity reasons, the quantization is suboptimal and some few bits are still unexploited. Residual quantization is a second layer of quantization which exploits these unused bits for refining the quantization error.

The below described embodiments of the present invention provide an encoder, a decoder
35 and methods which optimize this residual quantization.

Fig. 1 shows a block diagram of an encoder 100 according to an embodiment of the present invention. The Encoder 100 comprises a quantization stage 102 (e.g., a first quantization stage), an entropy encoder 104, a residual quantization stage 106 (e.g., a second quantization stage) and a coded signal former 108. The quantization stage 102 is configured to quantize an input signal 140 using a dead zone in order to obtain a plurality of quantized values 142 (q). The entropy encoder 104 is configured to encode the plurality of quantized values 142 (q) using an entropy encoding scheme in order to obtain a plurality of entropy encoded values 144 (e). The residual quantization stage 106 is configured to quantize a residual signal caused by a quantization in the quantization stage 102, wherein the residual quantization stage 106 is configured to determine at least one quantized residual value 146 (q_r) in dependence on the dead zone of the quantization stage 102. The coded signal former 108 is configured to form a coded signal 148 from the plurality of entropy encoded values 144 (e) and the at least one quantized residual value 146 (q_r).

The idea of the present invention is to reduce or even optimize the error between the (original) input signal and an inverse quantized version of a quantized version of the input signal by a residual quantization stage at the encoder side that takes into account the dead zone which was used for quantizing the input signal and an inverse quantization stage at the decoder side that also takes into account this dead zone when refining the inverse quantization level used for obtaining the inverse quantized signal.

In embodiments, the quantization stage 102 can be configured to perform a dead zone uniform threshold scalar quantization (DZ-UTSQ).

In embodiments, the coded signal former 108 can be configured to form the coded signal 148 by appending the at least one quantized residual value 146 or a plurality of quantized residual values 146 to the plurality of entropy encoded values 144 until the coded signal 148 comprises a maximum length available for a transfer to a decoder. It is not restricted that the bit-stream contains other information as scale factors defining the first quantization stage noise shaping, or prediction coefficients used for shaping the quantization noise and used in a post-filtering of the output signal in tie domain.

For example, the coded signal former 108 may be configured to provide a bit stream as the coded signal 148. Thereby, the coded signal former 108, for example, a multiplexer, can be configured to add at an end of the bit stream the at least one quantized residual value 146 or a plurality of quantized residual values 146. The bit stream generated by the encoder 100 may be transferred (e.g., transmitted or broadcasted) to a decoder or may be stored, for

example, on a non-volatile storage medium, for a later decoding by a decoder. Thereby, the bit stream may be transmitted or stored using data frames or data packets, wherein the bit stream may have to have a constant size (also referred herein as target bits) per data frame or data packet.

5

For obtaining a bit stream having a constant size or a predefined number of target bits the coded signal former 108 may be configured to append quantized residual values 146 to the entropy encoded values 144 until the bit stream reaches the predefined number of target bits. The residual quantization stage 106 may stop determining quantized residual values

10 146 when the bit stream comprises the predefined length or number of target bits.

In embodiments, the input signal 140 can be a frequency domain input signal 140. The encoder 100 can comprise a transformation unit configured to transform a time domain input signal into the frequency domain input signal 140.

15

Fig. 2 shows a block diagram of a decoder 120 according to an embodiment of the present invention. The decoder 120 comprises a coded signal parser 122, an entropy decoder 124 and an inverse quantization stage 126. The coded signal parser 122 is configured to parse a coded signal 148 in order to obtain a plurality of entropy encoded values 144 (e) and at least one quantized residual value 146 (q_r). The entropy decoder 124 is configured to decode the plurality of entropy encoded values 144 (e) using an entropy decoding scheme in order to obtain a plurality of quantized values 142 (q). The inverse quantization stage 126 is configured to inverse quantize the plurality of quantized values 142 (q) in order to obtain an output signal 150. Thereby, the inverse quantization stage 126 is configured to refine an

20 inverse quantization level used for obtaining the output signal 150 in dependence on the quantized residual value 146 (q_r) and a dead zone used in an encoder 100 in a quantization stage 106 for obtaining the plurality of quantized values 142 (q).

25

In embodiments, the inverse quantization stage 126 can be configured to refine the inverse quantization level by determining a refined inverse quantization level in dependence on the dead zone.

30

For example, the inverse quantization stage 126 may be configured to determine in dependence on the dead zone, or more precisely, in dependence on a width of the dead zone, a level by which the inverse quantization level has to be refined, i.e. increased or decreased, in order to obtain the refined inverse quantization level. Further, the inverse quantization stage 126 may be configured to determine at least two new inverse quantization

35

levels in dependence on the dead zone, and to obtain the output signal 150 by using one out of the at least two refined inverse quantization levels indicated by quantized residual value 146. In other words, the quantized residual value 146 indicates which out of the at least two refined inverse quantization levels is to be used for obtaining the output signal 150.

5

Fig. 3 shows a block diagram of a system 130 according to an embodiment of the present invention. The system 130 comprises the encoder 100 shown in Fig. 1 and the decoder 120 shown in Fig. 2.

10 In the following, features of the encoder 100 and decoder 120 and the coaction or interaction of features of the encoder 100 and decoder 120 are described in further detail.

Fig. 4 shows a block diagram of the residual quantization stage 106 according to an embodiment. The residual quantization stage 106 may comprise a residual quantizer 106',
15 an inverse quantizer 160 and a comparer 162. The inverse quantizer 160 can be configured to inverse quantize the plurality of quantized values 142 (q) provided by the quantization stage 102 in order to obtain an inverse quantized input signal 152 (x_q). The comparison stage 162 can be configured to compare the input signal 140 (x) and the inverse quantized input signal 152 (x_q) in order to obtain the residual signal 154. The residual quantizer 106'
20 can be configured to quantize the residual signal caused by the quantization stage 102.

In other words, the residual quantization block diagram is illustrated in Fig. 4. The spectrum 142 (q) is inversely quantized and compared to the original spectrum 140 (x). A second layer of quantization is then performed depending of the remaining bits available. The second
25 quantization step performed by the residual quantization stage 106 is usually a greedy quantization, i.e. that the quantization is performed line per line and each re-quantized value is done independently from the following transmitted information. In that way the residual quantization bit-stream 146 (q_r) can be truncated whenever the bit-stream 148 provided by the coded signal former 108 reaches the desired size.

30

As shown in Fig. 4, the residual quantization stage 106 may further comprise a control unit 164, e.g., an adjuster. The control unit 164 may be configured to control or optimize the residual quantizer 106'.

35 For example, the control unit 164 may be configured to control the residual quantizer 106' such that the residual quantizer 106' quantizes the residual signal 154 in dependence on the dead zone, or more precisely, in dependence on a width of the dead zone used in the

quantization stage 102 for obtaining the plurality of quantized values 142 (q). Further, the control unit 164 may be configured to control the residual quantizer 106' in dependence on a number of target bits and a number of consumed bits (e.g., consumed by the entropy encoded values 144 provided by the entropy encoder or by the entropy encoded values 144 and the quantized residual value(s) already provided by the residual quantizer 106'). Further, the control unit 164 may be configured to control the residual quantizer 106' in dependence on an information provided by the inverse quantizer 160. The information provided by the inverse quantizer 160 may include the width of the dead zone, which can be either fixed and modified adaptively, and may also include a scale factor applied in the first quantization stage for normalizing the spectrum and defining the quantization step, and may also include an indication if the quantized value was zeroed or not.

In a conventional residual quantization, Q_r performed by the residual quantization stage is a simple uniform scalar quantization of the difference $x[i]-x_{q[i]}$:

if $x[i] > x_{q[i]}$

$Q_r[i] = (\text{int})(0.5 + (x[i] - x_{q[i]}) / \Delta_r)$

Else

$Q_r[i] = (\text{int})(-0.5 + (x[i] - x_{q[i]}) / \Delta_r)$

wherein $x[i]$ is the input signal 140, wherein $x_{q[i]}$ is the inverse quantized input signal 152, wherein (int) is an integer rounding function, and wherein Δ_r is the quantization step of the residual quantizer Q_r which is usually smaller than the quantization step Δ used in the first quantizer Q . In general:

$\Delta_r = 0.5 \cdot \Delta$

Embodiments of the present invention solve two problems related to the residual quantization. The first and main problem is how to get the optimal Q_r (function of the residual quantization stage 106) knowing the first quantization stage 102. The second problem is how to minimize the mismatch between the encoder local synthesis and the decoder synthesis, when the number of remaining bits has to be estimated.

Through rate-distortion performance analysis, it is known that the uniform quantization (as used in a conventional residual quantization) is only optimal for memoryless and uniformly distributed sources. If an entropy coding is used afterwards, the uniform quantization is quasi optimal for a Gaussian source and at very high bit-rates. At lower rates the near optimal solution is to have a dead zone with uniform threshold scalar quantization (DZ-UTSQ). This family of quantizers is quasi optimal for a large range of distributions, e.g. Gaussian, Laplacian and generalized Laplacian. The dead zone factor can be optimized by different methods. It can be optimized in real time depending on estimate of the distribution. Simpler it can be fixed to a default best value found for expected input signals or adapted depending on some measures, like the tonality of the spectrum, which reflects also the distribution.

In the following, a solution to optimize the residual quantization Q_r performed by the residual quantization stage 106 depending on a first stage DZ-UTSQ 102 is presented. The dead zone parameter is called dz and DZ-UTSQ 102 is defined as:

if $x[i] > 0$

$Q[i] = (\text{int})(\text{rounding_dz} + (x[i])/\text{delta})$

Else

$Q[i] = (\text{int})(-\text{rounding_dz} + (x[i])/\text{delta})$

and

$x_q[i] = \text{delta} * Q[i]$

wherein $x[i]$ is the input signal 140, wherein $x_Q[i]$ is the inverse quantized input signal 152, wherein (int) is an integer rounding function, and wherein delta is the quantization step used in DZ-UTSQ 102, and wherein $\text{rounding_dz} = 1 - dz/2$.

Fig. 5 illustrates the DZ-UTSQ 102 scheme, where the scale is normalized by delta . The dead zone is usually larger than the normalized cell size of step 1. A dead zone of 1.25 is a good estimate for most of the frequency transformed audio samples. It can be reduced if the signal is noisier and enlarged when it is more tonal.

Embodiments of the present invention define the optimal quantization refinement of the error $x[i]-x_q[i]$. Because the residual coding is not entropy constrained no additional dead zone is adopted in the residual quantization Q_r . Moreover, the distribution of the quantization error of the first quantization stage 102 is assumed to be uniform in left and right parts of the quantization cell delimited by the reconstruction level 170. It is a high-rate assumption, i.e. the size of the new quantization cells are considered small enough for discarding non-even distributed errors within the cell. The assumption is valid for most of the target bit-rates.

There are two main cases: a sample was quantized with a non-zero value and a sample was quantized with a zero value.

For a non-zero quantized value, 1 bit can be allocated for the residual quantization Q_r per sample and define two relative reconstruction levels fac_m and fac_p :

$$fac_p = 0.5 - \text{rounding_dz} * 0.5 = 0.25 * (dz)$$

$$fac_m = 0.5 * \text{rounding_dz} = 0.5 * (1 - 0.5 * dz)$$

Thereby, fac_p may indicate a normalized absolute value by which a normalized absolute value of the inverse quantization level (or reconstruction level) 172 is to be increased in order to obtain a first refined inverse quantization level 174 of the two refined inverse quantization levels 174 and 176, wherein fac_m indicates a normalized absolute value by which the normalized absolute value of the inverse quantization level 172 is to be decreased in order to obtain a second refined inverse quantization level 176 of the two refined inverse quantization levels 174 and 176, and wherein dz is a normalized width of the dead zone, as will become clear from Fig. 6

Fig. 6 illustrates the two relative (or refined) reconstructive levels 174 and 176 for a reconstruction level 172 of 1. With 1 bit additional bit, the reconstruction level 172 can be refined with $1-fac_m$ (leading to the second refined inverse quantization level 176) or with $1+fac_p$ (leading to the first refined inverse quantization level 174). The original cell is split into two non-uniform cells. Because the quantization error of Q (quantization function of the first quantization stage 102) is assumed to be uniformly distributed within the new cells, the residual quantization Q_r is optimal in terms of R-D performance. Please note that the quantization Q and the residual quantization Q_r form an embedded quantization, i.e. the bit allocated to the residual quantization Q_r can be discarded and still $Q-1$ can be performed.

The residual quantization Qr performed by the residual quantization stage 106 can be summarized by:

$$prm[n] = \begin{cases} 0 & \text{if } x[i] < x_Q[i] \\ 1 & \text{otherwise} \end{cases}$$

5

wherein prm is a bit stream generated by the residual quantization stage 106 using the quantized residual value, wherein x[i] is the input signal, wherein x_Q[i] is the inverse quantized input signal, wherein n is an index that is incremented by 1 for each non-zero quantized value which is refined by Qr, and wherein i is an index that is incremented by 1 for each obtained quantized value.

10

The inverse Qr can be then expressed as:

$$\begin{aligned} & \text{if } (x_Q[i] > 0 \ \& \ n < N_{bits}) \text{ then} \\ x_Q[i] &= \begin{cases} x_Q[i] - \text{delta} * \text{fac_m} & \text{if } prm[n] = 0 \\ x_Q[i] + \text{delta} * \text{fac_p} & \text{otherwise} \end{cases} \\ & \text{else if } (n < N_{bits}) \text{ then} \\ x_Q[i] &= \begin{cases} x_Q[i] - \text{delta} * \text{fac_p} & \text{if } prm[n] = 0 \\ x_Q[i] + \text{delta} * \text{fac_m} & \text{otherwise} \end{cases} \end{aligned}$$

15

It can be seen that the inverse Qr is only performed for N_{bits} first bits. That means that the encoder can generate more bits than the encoder or decoder will actually decode. This mechanism is used when the remaining number of bits is estimated and when the local synthesis at the encoder side needs to be generated. The expected reconstructed signal is generated at the encoder although it is possible for the decoder to decode more or less bits depending of the true remaining available bits in the bit stream.

20

Alternatively, more than 1 bit can be allocated per sample to Qr. With the same principle the optimal reconstruction levels for 2 power of bits Qr reconstruction levels can be defined.

25

For a zero quantized value, the residual quantization Qr can be allocated with more than 1 bit. The reason is that for perceptual reason it is required to have zero as a reconstruction level. It avoids for example creating an artificial noisy signal during silence. A special 3 levels variable length code can be used:

30

0: code a zero

10: a negative reconstruction level

5 11: a positive reconstruction level

A new relative reconstruction level is computed, fac_z :

$fac_z = dz/3$

10

Thereby fac_z may indicate a normalized absolute value by which a normalized absolute value of the inverse quantization level 172 is to be increased in order to obtain a first refined inverse quantization level 174 of the two refined inverse quantization levels 174 and 176 and a normalized absolute value by which a normalized absolute value of the inverse

15 quantization level is to be decreased in order to obtain a second refined inverse quantization level 176 of the two refined inverse quantization levels 174 and 176, and wherein dz is a normalized width of the dead zone, as will become clear from Fig. 7.

Fig. 7 illustrates the residual quantization Q_r performed by the residual quantization stage

20 106 for a zero quantized values 142. The cell around zero is divided in three uniform new cells.

The residual quantization Q_r performed by the residual quantization stage 106 for a zero quantized value can be summarized by

25

$$prm[n] = \begin{cases} 0 & \text{if } |x[i]| < (C \cdot x_Q[i]) \\ 1 & \text{otherwise} \end{cases}$$

if $prm[n] = 1$ then

$$prm[n+1] = \begin{cases} 0 & \text{if } x[i] < 0 \\ 1 & \text{otherwise} \end{cases}$$

wherein C depends on the dead zone of the quantization stage and may be calculated to

30 $C = \delta \cdot (fac_z/2)$, wherein prm is a bit stream generated by the residual quantization stage 106 using the quantized residual value, wherein $x[i]$ is the input signal, wherein $x_Q[i]$ is the inverse quantized input signal. The index n is incremented by 1 for each zero quantized

value requantized to zero, wherein n is incremented by 2 for each zero quantized value requantized to a non-zero value.

The inverse Qr can be then expressed as:

5

$$\text{if } (n < N_{bits}) \text{ then}$$

$$x_Q[i] = \begin{cases} \text{delta} * \text{fac_z} & \text{if } \text{prm}[n] = 1 \ \& \ \text{prm}[n+1] = 1 \\ -\text{delta} * \text{fac_z} & \text{if } \text{prm}[n] = 1 \ \& \ \text{prm}[n+1] = 0 \\ 0 & \text{otherwise} \end{cases}$$

10

Embodiments of the present invention can easily be extended with the assumption that the distribution within the original quantization cell is not uniform. In this case, the relative reconstruction levels can be derived depending on the distribution of the quantization error. A way of achieving it is to split the original quantization cell into non-uniform new smaller cells. A second dead zone parameter can be used as well.

15

In the following, further embodiments of the encoder 100 and decoder 120 are briefly described.

First, the encoder 100 is described.

20

The residual quantization is a refinement quantization layer refining the first SQ stage (or quantization stage 102). It exploits eventual unused bits, i.e. unused bits = target_bits-nbbits, where nbbits is the number of bits consumed by the entropy coder 104. The residual quantization adopts a greedy strategy and no entropy in order to stop the coding whenever the bit stream reaches the desired size.

25

The refinement consists of re-quantizing the quantized spectrum line per line. First, the non-zero quantized lines are processed with a 1 bit residual quantizer:

if ($X[k] < \hat{X}[k]$) *then*
 write_bit(0)
else then
 write_bit(1)

30

Thereby, $X[k]$ is a scaled sample of the input signal 140 and $\hat{X}[k]$ is the scaled corresponding sample of the inverse quantized input signal 152.

Finally, if remaining bits allow, the zero quantized lines are considered and quantized with on 3 levels as follows:

```

fac_z = (1 - rounding_dz) · 0.66
if( |X[k] | < 0.5 · fac_z ·  $\hat{X}[k]$  ) then
5   write_bit(0)
else then
   write_bit(1)
   write_bit((1 + sgn(X[k])) / 2)

```

Thereby, $X[k]$ is a scaled sample of the input signal 140, $X[\hat{k}]$ is the corresponding scaled sample of the inverse quantized input signal 152, *fac_z* may indicate a normalized absolute value by which a normalized absolute value of the inverse quantization level 172 is to be increased in order to obtain a first refined inverse quantization level 174 of the two refined inverse quantization levels 174 and 176 and a normalized absolute value by which a normalized absolute value of the inverse quantization level is to be decreased in order to obtain a second refined inverse quantization level 176 of the two refined inverse quantization levels 174 and 176, wherein *rounding_dz*=1-*dz*/2.

Second, the decoder 120 is described.

The remaining bits refine the non-zero decoded lines. 1 bit per non-zero spectral value is read:

```

fac_p = (1 - rounding_dz) / 2
fac_m = rounding_dz / 2
if (read_bit() == 0) then
 $\hat{X}[k] = \begin{cases} \hat{X}[k] - \textit{fac\_m} & \text{if } \hat{X}[k] > 0 \\ \hat{X}[k] - \textit{fac\_p} & \text{otherwise} \end{cases}$ 
else then
 $\hat{X}[k] = \begin{cases} \hat{X}[k] + \textit{fac\_p} & \text{if } \hat{X}[k] > 0 \\ \hat{X}[k] + \textit{fac\_m} & \text{otherwise} \end{cases}$ 

```

Thereby, $X[k]$ is the input signal 140, $X[\hat{k}]$ is the inverse quantized input signal 152, *fac_p* may indicate a normalized absolute value by which a normalized absolute value of the

inverse quantization level (or reconstruction level) 172 is to be increased in order to obtain a first refined inverse quantization level 174 of the two refined inverse quantization levels 174 and 176, and *fac_m* may indicate a normalized absolute value by which the normalized absolute value of the inverse quantization level 172 is to be decreased in order to obtain a second refined inverse quantization level 176 of the two refined inverse quantization levels 174 and 176, wherein $\text{rounding_dz} = 1 - \text{dz}/2$

If at least 2 bits are left to read, a zero value is refined as:

$$\text{fac_z} = (1 - \text{rounding_dz}) \cdot 0.66$$

if(*read_bit*() == 0) *then*

$$\hat{X}[k] = 0$$

else then

if(*read_bit*() == 0) *then*

$$\hat{X}[k] = -\text{fac_z}$$

else then

$$\hat{X}[k] = \text{fac_z}$$

Thereby, $X[k]$ is a scaled sample of the input signal 140, $X[\hat{k}]$ is the corresponding scaled sample of the inverse quantized input signal 152, *fac_z* may indicate a normalized absolute value by which a normalized absolute value of the inverse quantization level 172 is to be increased in order to obtain a first refined inverse quantization level 174 of the two refined inverse quantization levels 174 and 176 and a normalized absolute value by which a normalized absolute value of the inverse quantization level is to be decreased in order to obtain a second refined inverse quantization level 176 of the two refined inverse quantization levels 174 and 176, wherein $\text{rounding_dz} = 1 - \text{dz}/2$.

Fig. 8 shows a flow chart of a method for encoding 200 according to an embodiment. The method 200 comprises a step 202 of quantizing an input signal in order to obtain a plurality of quantized values using a dead zone; a step 204 of encoding the plurality of quantized values using an entropy encoding scheme in order to obtain a plurality of entropy encoded values; a step 206 of quantizing a residual signal caused by a quantization by the quantization stage and determining a plurality of quantized residual values in dependence on the dead zone of the quantization stage; and a step 208 of forming a bit stream from the plurality of entropy encoded values and the plurality of quantized residual values.

Fig. 9 shows a flow chart of a method for decoding 220 according to an embodiment. The method 220 comprises a step 222 of parsing a coded signal in order to obtain a plurality of entropy encoded values and a quantized residual value; a step 224 of decoding the plurality of entropy encoded values using an entropy decoding scheme in order to obtain a plurality of quantized values; a step 226 of inverse quantizing the plurality of quantized values using a dead zone in order to obtain an output signal; and a step 228 of refining an inverse quantization level used for obtaining the output signal in dependence on a dead zone and the quantized residual value.

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. Some or all of the method steps may be executed by (or using) a hardware apparatus, like for example, a microprocessor, a programmable computer or an electronic circuit. In some embodiments, some one or more of the most important method steps may be executed by such an 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 Blu-Ray, 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. Therefore, the digital storage medium may be computer readable.

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.

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.

5

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. The data carrier, the digital storage medium or the recorded medium are typically tangible and/or non-

10

transitory.

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

15

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.

20

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

A further embodiment according to the invention comprises an apparatus or a system configured to transfer (for example, electronically or optically) a computer program for performing one of the methods described herein to a receiver. The receiver may, for example, be a computer, a mobile device, a memory device or the like. The apparatus or system may, for example, comprise a file server for transferring the computer program to the receiver .

25

30

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.

35

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.

5

Claims

1. Encoder, comprising:

5 a quantization stage configured to quantize an input signal using a dead zone in order to obtain a plurality of quantized values;

an entropy encoder configured to encode the plurality of quantized values using an entropy encoding scheme in order to obtain a plurality of entropy encoded values;

10

a residual quantization stage configured to quantize a residual signal caused by the quantization stage, wherein the residual quantization stage is configured to determine, for a non-zero quantized value, at least one quantized residual value in dependence on a width of the dead zone of the quantization stage; and

15

a coded signal former configured to form a coded signal from the plurality of entropy encoded values and the at least one quantized residual value;

20

wherein the coded signal former is configured to form the coded signal by appending the at least one quantized residual value or a plurality of quantized residual values to the plurality of entropy encoded values until the coded signal comprises a maximum length available for a transfer to a decoder;

25

wherein the residual quantization stage is configured to determine the at least one quantized residual value such that the quantized residual value comprises 1 bit for a non-zero quantized value, wherein the residual quantization stage is configured to determine the quantized residual value based on the comparison

$$x[i] < x_Q[i]$$

30

wherein $x[i]$ is the input signal, wherein $x_Q[i]$ is the inverse quantized input signal, and wherein i is an index that is incremented by 1 for each obtained quantized value;

wherein the residual quantization stage is configured to determine the quantized residual value such that the quantized residual value comprises 2 bits for a zero quantized value, wherein the residual quantization stage is configured to determine the quantized residual value based on the comparisons

5

$$|x[i]| < (C \cdot x_Q[i]) \text{ and } x[i] < 0$$

wherein C depends on the width of the dead zone of the quantization stage, wherein $x[i]$ is the input signal, wherein $x_Q[i]$ is the inverse quantized input signal, and wherein i is an index that is incremented by 1 for each obtained quantized value;

10

wherein the at least one quantized residual value is not entropy encoded.

2. Encoder according to claim 1, wherein the residual quantization stage comprises:

15

an inverse quantizer configured to inverse quantize the plurality of quantized values in dependence on the dead zone of the quantization stage in order to obtain an inverse quantized input signal;

20

wherein the residual quantization stage is configured to determine the at least one quantized residual value such that the quantized residual value comprises 1 bit for a non-zero quantized value; and

25

wherein the residual quantization stage is configured to determine, for the non-zero quantized value, that the quantized residual value comprises a logic 0 if the input signal is smaller than the inverse quantized input signal for the non-zero quantized value, and to determine, for the non-zero quantized value, that the quantized residual value comprises a logic 1 else.

30

3. Encoder according to one of the claims 1 to 2, wherein the residual quantization stage comprises:

an inverse quantizer configured to inverse quantize the plurality of quantized values in dependence on the dead zone in order to obtain an inverse quantized input signal; and

35

a comparer configured to compare the input signal and the inverse quantized input signal.

4. Encoder according to claim 3, wherein the comparer is configured to compare the input signal and the inverse quantized input signal in order to obtain the residual signal;

wherein the residual quantization stage is configured to quantize the residual signal in dependence on the dead zone.

5. Encoder according to claim 4, wherein the residual quantization stage is configured to determine the at least one quantized residual value such that the quantized residual value comprises 1 bit for a non-zero quantized value; and

wherein the residual quantization stage is configured to determine, for the non-zero quantized value, that the quantized residual value comprises a logic 0 if the residual signal is negative for the non-zero quantized value, and to determine, for the non-zero quantized value, that the quantized residual value comprises a logic 1 else.

6. Encoder according to one of the claims 1 to 5, wherein the residual quantization stage is configured to determine the at least one quantized residual value such that the quantized residual value comprises 1 bit for a non-zero quantized value, wherein the residual quantization stage is configured to determine the quantized residual value based on the syntax

$$prm[n] = \begin{cases} 0 & \text{if } x[i] < x_Q[i] \\ 1 & \text{otherwise} \end{cases}$$

wherein prm is a bit stream generated by the residual quantization stage using the quantized residual value, wherein x[i] is the input signal, wherein x_Q[i] is the inverse quantized input signal, wherein n is an index that is incremented by 1 for each non-zero quantized value, and wherein i is an index that is incremented by 1 for each obtained quantized value.

7. Encoder according to one of the claims 1 to 6, wherein the residual quantization stage is configured to determine the quantized residual value such that the quantized residual value comprises 2 bits for a zero quantized value,

wherein the residual quantization stage is configured to determine the quantized residual value based on the syntax

$$prm[n] = \begin{cases} 0 & \text{if } |x[i]| < (C \cdot x_Q[i]) \\ 1 & \text{otherwise} \end{cases}$$

5 *if* $prm[n] == 1$ *then*

$$prm[n + 1] = \begin{cases} 0 & \text{if } x[i] < 0 \\ 1 & \text{otherwise} \end{cases}$$

10 wherein C depends on the dead zone of the quantization stage, wherein prm is a bit stream generated by the residual quantization stage using the quantized residual value, wherein $x[i]$ is the input signal, wherein $x_Q[i]$ is the inverse quantized input signal, wherein n is an index that is incremented by 1 for each zero quantized value that is requantized to a zero quantized value and incremented by 2 for each zero quantized value that is requantized to a non-zero quantized value, and wherein i is an index that is incremented by 1 for each obtained quantized value.

15

8. Encoder according to one of the claims 1 to 7, wherein the coded signal former is configured to provide a bit stream as the coded signal, wherein the coded signal former is configured to form the bit stream from the plurality of entropy encoded values and the plurality of quantized residual values, wherein the coded signal former is configured to append the quantized residual values to the entropy encoded values, wherein the residual quantization stage comprises:
- 20

a residual quantizer; and

25

an adjuster configured to control the residual quantizer to quantize the residual signal in dependence on a width of the dead zone used in the quantization stage for obtaining the plurality of quantized values;

30

wherein the adjuster is configured to obtain a number of target bits and a number of consumed bits; and

wherein the adjuster is configured to control the residual quantization stage to stop determining quantized residual values when the bit stream comprises the number of target bits.

5 9. Decoder, comprising:

a coded signal parser configured to parse a coded signal in order to obtain a plurality of entropy encoded values and at least one quantized residual value;

10 an entropy decoder configured to decode the plurality of entropy encoded values using an entropy decoding scheme in order to obtain a plurality of quantized values; and

15 an inverse quantization stage configured to inverse quantize the plurality of quantized values in order to obtain an output signal;

wherein the inverse quantization stage is configured to refine an inverse quantization level used for obtaining the output signal in dependence on the quantized residual value;

20

wherein the inverse quantization stage is configured to refine an inverse quantization level for a non-zero quantized value in dependence on a quantized residual value and a width of the dead zone.

25 10. Decoder according to claim 9, wherein the inverse quantization stage is configured to refine the inverse quantization level by determining a refined inverse quantization level in dependence on the dead zone.

30 11. Decoder according to claim 10, wherein the inverse quantization stage is configured to determine two refined inverse quantization levels for a non-zero quantized value, wherein the inverse quantization stage is configured to obtain the output signal by using one out of the two refined inverse quantization levels indicated by quantized residual value.

35 12. Decoder according to claim 11, wherein the inverse quantization stage is configured to increase a normalized absolute value of the inverse quantization level

by an increase value in order to obtain a first of the two refined inverse quantization levels;

wherein the inverse quantization stage is configured to decrease a normalized absolute value of the inverse quantization level by a decrease value in order to obtain a second of the two refined inverse quantization levels ; and

wherein the increase value and the decrease value are different from each other.

- 10 13. Decoder according to one of the claims 11 or 12, wherein the inverse quantization stage is configured to determine the two refined inverse quantization levels for a non-zero quantized value based on the two factors:

15
$$\text{fac_p} = 0,25 * \text{dz}$$

$$\text{fac_m} = 0,5 * (1 - 0,5 * \text{dz})$$

20 wherein fac_p indicates a normalized absolute value by which a normalized absolute value of the inverse quantization level is to be increased in order to obtain a first of the two refined inverse quantization levels, wherein fac_m indicates a normalized absolute value by which the normalized absolute value of the inverse quantization level is to be decreased in order to obtain a second of the two refined inverse quantization levels, and wherein dz is a normalized width of the dead zone.

- 25 14. Decoder according to one of the claims 9 to 13, wherein the inverse quantization stage is configured to determine two refined inverse quantization levels for a zero quantized value, wherein the inverse quantization stage is configured to obtain the output signal by using one out of the inverse quantization level and two refined inverse quantization levels indicated by quantized residual value.

30

15. Decoder according to claim 14, wherein the inverse quantization stage is configured to increase a normalized absolute value of the inverse quantization level by an increase value in order to obtain a first of the two refined inverse quantization levels;

wherein the inverse quantization stage is configured to decrease a normalized absolute value of the inverse quantization level by a decrease value in order to obtain a second of the two refined inverse quantization levels.

- 5 16. Decoder according to one of the claims 14 or 15, wherein the inverse quantization stage is configured to determine the two refined inverse quantization levels for the zero quantized value based on the factor:

$$\text{fac_z} = \text{dz} / 3$$

10

wherein fac_z indicates a normalized absolute value by which a normalized absolute value of the inverse quantization level is to be increased in order to obtain a first of the two refined inverse quantization levels and a normalized absolute value by which a normalized absolute value of the inverse quantization level is to be decreased in order to obtain a second of the two refined inverse quantization levels, and wherein dz is a normalized width of the dead zone.

15

17. System, comprising:

20

an encoder according to one of the claims 1 to 8; and

a decoder according to one of the claims 9 to 16.

18. Method for encoding, the method comprising:

25

quantizing an input signal in order to obtain a plurality of quantized values using a dead zone;

encoding the plurality of quantized values using an entropy encoding scheme in order to obtain a plurality of entropy encoded values;

30

quantizing a residual signal caused by quantizing the input signal, wherein quantizing the residual signal comprises determining, for a non-zero quantized value, at least one quantized residual value in dependence on a width of the dead zone of the quantization stage; and

35

forming a bit stream from the plurality of entropy encoded values and the plurality of quantized residual values;

5 wherein forming the bitstream comprises appending the at least one quantized residual value or a plurality of quantized residual values to the plurality of entropy encoded values until the coded signal comprises a maximum length available for a transfer to a decoder;

10 wherein quantizing the residual signal comprises determining the at least one quantized residual value such that the quantized residual value comprises 1 bit for a non-zero quantized value, wherein the residual quantization stage is configured to determine the quantized residual value based on the comparison

$$0 \text{ if } x[i] < x_Q[i]$$

15 wherein $x[i]$ is the input signal, wherein $x_Q[i]$ is the inverse quantized input signal, and wherein i is an index that is incremented by 1 for each obtained quantized value;

20 wherein quantizing the residual signal comprises determining the quantized residual value such that the quantized residual value comprises 2 bits for a zero quantized value, wherein the residual quantization stage is configured to determine the quantized residual value based on the comparisons

$$|x[i]| < (C \cdot x_Q[i]) \text{ and } x[i] < 0$$

25 wherein C depends on the width of the dead zone of the quantization stage, wherein $x[i]$ is the input signal, wherein $x_Q[i]$ is the inverse quantized input signal, and wherein i is an index that is incremented by 1 for each obtained quantized value;

30 wherein the at least one quantized residual value is not entropy encoded.

19. Method for decoding, the method comprising:

parsing a coded signal in order to obtain a plurality of entropy encoded values and a quantized residual value;

- decoding the plurality of entropy encoded values using an entropy decoding scheme in order to obtain a plurality of quantized values;
- 5 inverse quantizing the plurality of quantized values in order to obtain an output signal;
and
- refining an inverse quantization level used for obtaining the output signal in dependence on a width of a dead zone and the quantized residual value;
- 10 wherein refining the inverse quantization level comprises refining an inverse quantization level for a non-zero quantized value in dependence on a quantized residual value and a width of the dead zone.
20. Computer program for performing a method according to claim 18 or 19.
- 15