Title: METHODS AND DEVICES FOR SCALABLE VIDEO CODING

Abstract: A method of encoding video data comprises encoding video data having a first resolution in conformity with HEVC (13) to obtain video data of a base layer (14) and decoding the base-layer video data in conformity with HEVC (15). The decoded base-layer video data is upsampled (16) to generate decoded video data having a second resolution higher than the first resolution. A difference is formed (17) between the generated decoded video data having the second resolution and further video data, having said second resolution and corresponding to the first-resolution video data, to generate data of a residual image. The residual-image data is compressed (18, 19) to generate video data of an enhancement layer (20). Such a method can be used to provide a UHD codec with low complexity based on scalable encoding.
METHODS AND DEVICES FOR SCALABLE VIDEO CODING

FIELD OF THE INVENTION

The present invention concerns methods for encoding and decoding an image, and associated apparatuses and programs.

The invention is particularly useful for the encoding of digital video sequences made of images or "frames".

BACKGROUND OF THE INVENTION

Video compression algorithms, such as those standardized by the standardization organizations ITU, ISO, and SMPTE, exploit the spatial and temporal redundancies of images in order to generate bitstreams of data of smaller size than original video sequences. These powerful video compression tools, known as spatial (or intra) and temporal (or inter) predictions, make the transmission and/or the storage of video sequences more efficient.

ITU-T and ISO/MPEG decided to launch in January 2010 a new video coding standard project, named High Efficiency Video Coding (HEVC).

The HEVC codec design is similar to that of most previous so-called block-based hybrid transform codecs such as H.263, H.264, MPEG-1, MPEG-2, MPEG-4, SVC. The HEVC codec uses the spatial and temporal redundancies of the images in order to generate data bitstreams of reduced size compared with these video sequences. Such compressions make the transmission and/or storage of the video sequences more effective.

Video encoders and/or decoders (codecs) are often embedded in portable devices with limited resources, such as cameras or camcorders. Conventional embedded codecs can process at best high definition (HD) digital videos, i.e. 1080x1920 pixel frames.

Real time encoding and decoding are however limited by the limited resources of the portable devices, especially regarding slow access to the working memory (e.g. random access memory, or RAM) and regarding the central processing unit (CPU).

This is particularly striking for the encoding or decoding of ultra-high definition (UHD) digital videos that are about to be handled by the latest cameras. This
is because the amount of pixel data to consider for spatial or temporal prediction is huge.

UHD is typically four times (4k2k pixels) the definition of an HD video which is the current standard definition video. Furthermore, very ultra high definition, which is sixteen times that definition (i.e. 8k4k pixels), is even being considered in a more long-term future.

Taking account of these constraints in terms of limited power and memory access bandwidth, it is desirable to provide a UHD codec with low complexity based on scalable encoding.

SUMMARY OF THE INVENTION

<1st to 4th aspects>
According to a 1st aspect of the present invention there is provided a method of encoding video data comprising:

- encoding video data having a first resolution in conformity with HEVC to obtain video data of a base layer; and
- decoding the base-layer video data in conformity with HEVC, upsampling the decoded base-layer video data to generate decoded video data having a second resolution higher than said first resolution, forming a difference between the generated decoded video data having said second resolution and further video data, having said second resolution and corresponding to said first-resolution video data, to generate data of a residual image, and compressing the residual-image data to generate video data of an enhancement layer.

Preferably, the compression of the residual-image data does not involve temporal prediction.

Preferably, the compression of the residual-image data does not involve spatial prediction.

Preferably, the compression of the residual data employs a method embodying one or more of the 5th, 9th, 11th and 15th aspects of the present invention described hereinafter.

In one embodiment, the compression of the residual-image data comprises applying a discrete cosine transformation (DCT) to obtain DCT coefficients.

Preferably the method further comprises employing a parametric probabilistic model of the DCT coefficients.
Such a parametric probabilistic model is preferably obtained for each type of DCT coefficient.

The DCT may be a block-based DCT, in which case such a parametric probabilistic model is preferably obtained for each DCT coefficient position within a DCT block.

In such a case also the method may further comprise fitting the parametric probabilistic model onto respective collocated DCT coefficients of some or all DCT blocks of the residual image.

In one embodiment, the compression of the residual-image data comprises employing the parametric probabilistic model to choose quantizers from a pool of available quantizers and quantizing the residual data using the chosen quantizers.

The available quantizers of said pool are preferably pre-computed quantizers dedicated to a parametric probabilistic model.

In another embodiment the compression of the residual-image data comprises entropy encoding of quantized symbols and employing a parametric probabilistic model to obtain a probabilistic distribution of possible symbols of an alphabet associated with each DCT coefficient, which alphabet is used for the entropy encoding.

The same parametric probabilistic model is preferably employed both for choosing quantizers and for entropy encoding.

The parametric probabilistic model is a Generalised Gaussian Distribution $GGD(\alpha, \beta)$ having a zero mean. This requires only two parameters, which makes it possible to supply the parameters to the decoder with low overhead.

Preferably, parameters of the parametric probabilistic model are determined in dependence upon one or more of:

- a video content;
- an index of the DCT coefficient within a DCT block;
- an encoding mode used for a collocated block of the base layer; and
- a size of block to encode.

Preferably, information about parameters of the parametric probabilistic model is supplied to a decoder.

In one embodiment, information about a set of saturated coefficients determined at the encoder to minimize a rate is supplied to a decoder.

In another embodiment, information about the chosen quantizers is supplied to a decoder.
Preferably, the quantizers are chosen based on a rate-distortion criterion.
According to a 2nd aspect of the present invention there is provided a method of decoding a scalable bitstream, comprising:

decoding, in conformity with HEVC, encoded video data of a base layer of the bitstream to obtain video data having a first resolution, and upsampling the first-resolution video data to generate video data having a second resolution higher than said first resolution; and

decoding compressed video data of an enhancement layer of the bitstream to obtain data of a residual image, and forming a sum of the generated second-resolution video data and the residual-image data to generate decoded video data having said second resolution.

Preferably, the decoding of the compressed video data of the enhancement layer does not involve temporal prediction.

Preferably, the decoding of the compressed video data of the enhancement layer does not involve spatial prediction.

Preferably, the decoding of the compressed video data of the enhancement layer employs a method embodying one or more of the 2nd, 6th, 13th, 17th and 20th aspects of the present invention described hereinafter.

In one embodiment, the compressed video data of the enhancement layer comprises encoded discrete cosine transformed (DCT) coefficients.

In such a case, preferably, the method further comprises employing a parametric probabilistic model of the DCT coefficients.

Preferably, such a parametric probabilistic model is obtained for each type of DCT coefficient.

The DCT may be a block-based DCT, in which case such a parametric probabilistic model is obtained for each DCT coefficient position within a DCT block.

In one embodiment, the decoding of the compressed video data of the enhancement layer comprises employing the parametric probabilistic model to choose quantizers from a pool of available quantizers and using the chosen quantizers for inverse quantization of the encoded DCT coefficients.

Preferably, the available quantizers of said pool are pre-computed quantizers dedicated to a parametric probabilistic model.

In another embodiment, the decoding of the compressed video data of the enhancement layer comprises entropy decoding of encoded quantized symbols obtained from the compressed video data to generate quantized DCT coefficients and
employing a parametric probabilistic model to obtain a probabilistic distribution of possible symbols of an alphabet associated with each DCT coefficient, which alphabet is used for the entropy decoding.

The same parametric probabilistic model is preferably employed both for choosing quantizers and for entropy decoding.

In one embodiment, the parametric probabilistic model is a Generalised Gaussian Distribution GGD(a, β) having a zero mean. Such a model has only two parameters which means that information about the parameters can be received from the encoder with low overhead.

Information about parameters of the parametric probabilistic model can be received from an encoder.

Alternatively, information about a set of saturated coefficients determined at an encoder to minimize a rate is received from the encoder.

In one embodiment, information about the chosen quantizers is received from an encoder.

The quantizers are preferably chosen based on a rate-distortion criterion.

According to a 3rd aspect of the present invention, there is provided apparatus for encoding video data comprising:

- means for encoding video data having a first resolution in conformity with HEVC to obtain video data of a base layer;
- means for decoding the base-layer video data in conformity with HEVC;
- means for upsampling the decoded base-layer video data to generate decoded video data having a second resolution higher than said first resolution;
- means for forming a difference between the generated decoded video data having said second resolution and further video data, having said second resolution and corresponding to said first-resolution video data, to generate data of a residual image; and
- means for compressing the residual-image data to generate video data of an enhancement layer.

Preferably, the means for compressing the residual data comprises apparatus embodying one or more of the 3rd, 7th, 10th, 12th and 16th aspects of the present invention described hereinafter.

According to a 4th aspect of the present invention, there is provided apparatus for decoding a scalable bitstream, comprising:
means for decoding, in conformity with HEVC, encoded video data of a
base layer of the bitstream to obtain video data having a first resolution;
means for upsampling the first-resolution video data to generate video data
having a second resolution higher than said first resolution;
means for decoding compressed video data of an enhancement layer of the
bitstream to obtain data of a residual image; and
means for forming a sum of the generated second-resolution video data
and the residual-image data to generate decoded video data having said second
resolution.

Preferably, the means for decoding of the compressed video data of the
enhancement layer comprises apparatus embodying one or more of the 4th, 8th, 14th,
18th and 21st aspects of the present invention described hereinafter.

<5th to 8th aspects>

5th to 8th aspects of the present invention, set out below, may be used for
the encoding of any image, regardless of the enhancement nature of the image, and
furthermore regardless of its resolution.

These aspects of the invention are particularly advantageous when
encoding images without prediction.

The features of the 5th to 8th aspects of the invention may be provided in
combination with the features of the 1st to 4th aspects but this is not essential and it is
possible to use the features of the 5th to 8th aspects independently of the features of the
1st to 4th aspects.

According to the 5th aspect of the present invention, there is provided a
method for encoding at least one block of pixels, the method comprising:

- transforming pixel values in the spatial domain for said block into a
  plurality of coefficients each having a coefficient type;
- determining a set of quantizers each associated with a corresponding
coefficient type such that an estimated quality parameter in the spatial domain,
expressed as a function of estimated distortions each associated with a quantizer of
the set, meets a predetermined criterion and such that the sum of rates each
associated with a quantizer of the set is minimised;
- quantizing at least one coefficient having a given coefficient type into a
  quantized symbol using the quantizer determined for said given coefficient type;
- encoding the quantized symbol.
Thus, although an optimal set of quantizers is determined for quantizing coefficients (in the frequency domain), optimisation is performed in view of a quality criterion in the spatial domain, where the image is eventually to be shown to the user.

The step of transforming pixel values corresponds for instance to a transformation from the spatial domain (pixels) to the frequency domain (e.g. into coefficients each corresponding to a specific spatial frequency). For example, the transforming step includes applying a block based Discrete Cosine Transform; each of said coefficient types may then correspond to a respective coefficient index or position.

Said estimated quality parameter depends for instance on the sum of the squared estimated distortions associated with quantizers of the set. Such a parameter is linked to the PSNR as explained below and is thus of particular interest.

In the embodiment described below, the method comprises a step of computing said pixel values by subtracting values obtained by decoding a base layer to values representing pixels of an image. In this application, the pixel values to be encoded thus correspond to residual values and form, once encoded, an enhancement layer. When the base layer uses at least two distinct encoding modes, a set of quantizers may be determined for each of said two encoding modes. As explained below, the optimal quantizer may vary greatly depending on the encoding mode concerned.

It may be provided that the step of determining the set of quantizers includes selecting, among optimal quantizers each associated with a rate and corresponding distortion, an optimal quantizer for each coefficient type such that the sum of the rates associated with selected quantizers is minimal and the estimated quality parameter resulting from the distortions associated with said selected quantizers corresponds to a predetermined target. Selection among optimal quantizers is an interesting practical solution to perform the necessary optimisation.

In the same respect, in accordance with the embodiment proposed here and described below, said selecting the optimal quantizers for each coefficient type may for instance be performed by minimisation under Karush-Kuhn-Tucker necessary conditions, possibly by a fixed point algorithm.

The method may also include, for at least one coefficient type, a step of determining a probabilistic model for coefficients of said at least one coefficient type based on a plurality of values of coefficients of said at least one coefficient type, wherein optimal quantizers, among which the optimal quantizer for said at least one coefficient type is selected, are determined based on said probabilistic model. The
optimisation process may thus be performed based on the probabilistic model identified for the concerned coefficient type, which is a convenient way to take into account effective values of the coefficient in the optimisation process.

Said probabilistic model is for instance one of a plurality of probabilistic models each defined by model parameters; parameters defining optimal quantizers may be stored in association with corresponding model parameters. Optimal quantizers are thus tabulated depending on the various probabilistic models that may be encountered. Distinct probabilistic models may be determined for the plurality of coefficient types. This improves flexibility in the optimisation process.

According to the embodiment proposed below, optimal quantizers, associated with a rate and distortion, and among which the optimal quantizer for said at least one coefficient type is selected, are determined off-line, i.e. in an apparatus possibly distinct from the encoding device and prior to the step of transforming pixel values. Parameters defining the resulting optimal quantizers may then be stored as mentioned above in the encoding device in order to perform the optimisation process.

A quantizer is for instance defined by at least one parameter taken among a number of quantization intervals, a limit value for a quantization interval and a centroid of a quantization interval.

Quantization and encoding may be performed for each and every coefficients resulting from the transforming step.

However, in the embodiment proposed here and described below, the method comprises a step of determining, for each coefficient type, an estimated value representative of a ratio between a distortion variation (e.g. here decrease) provided by encoding a coefficient having the concerned type and a rate increase resulting from encoding said coefficient; the step of quantizing coefficients may then be applied only to a subset of said plurality of coefficients (i.e. the coefficients subjected to the quantization step form such a subset) and the estimated values for coefficient types of coefficients in said subset may then be larger than the highest estimated value over coefficient types of coefficients not included in said subset.

Thanks to the estimated value or ratio for each coefficient type, it is possible to order the various coefficient types by decreasing estimated value, i.e. by decreasing merit of encoding as explained below, and the encoding process may be applied only to coefficients having the higher values (i.e. ratios), forming the subset defined above.
The step of determining the set of quantizers may include a step of
determining sets of quantizers each associated with a possible subset of said plurality
of coefficients and a step of selecting, among said sets of quantizers, a set for which
the sum of rates each associated with a quantizer of the set is minimal, thus selecting
an associated subset. The number of coefficients to be quantized and encoded is thus
determined during the optimisation process. A plurality of possible subsets are
considered; however, as the coefficient types are ordered by decreasing encoding
merit, only N+1 subsets need be considered if N is the total number of coefficients.

According to the 6th aspect of the present invention, there is provided a
method for decoding a set of data representing at least one block of pixels, the method
comprising:
- decoding said data into a plurality of symbols each corresponding to a
coefficient type;
- determining a set of quantizers each associated with a corresponding
coefficient type such that an estimated quality parameter in the spatial domain,
expressed as a function of estimated distortions each associated with a quantizer of
the set, meets a predetermined criterion and such that the sum of rates each
associated with a quantizer of the set is minimised;
- dequantizing at last one symbol corresponding to a given coefficient type
into a dequantized coefficient using the quantizer determined for said given coefficient
type;
- transforming a plurality of dequantized coefficient into pixel values in the
spatial domain for said block.

The step of determining the set of quantizers may include estimating
distortion for a coefficient type based on a parameter defining a distribution of the
values of coefficients of said coefficient type and included in the set of data.

According to the 7th aspect of the present invention, there is provided
apparatus for encoding at least one block of pixels, comprising:
- means for transforming pixel values in the spatial domain for said block
into a plurality of coefficients each having a coefficient type;
- means for determining a set of quantizers each associated with a
corresponding coefficient type such that an estimated quality parameter in the spatial
domain, expressed as a function of estimated distortions each associated with a
quantizer of the set, meets a predetermined criterion and such that the sum of rates
each associated with a quantizer of the set is minimised;
- means for quantizing at least one coefficient having a given coefficient type into a quantized symbol using the quantizer determined for said given coefficient type;

- means for encoding the quantized symbol.

According to the 8th aspect of the present invention, there is provided apparatus for decoding a set of data representing at least one block of pixels, comprising:

- means for decoding said data into a plurality of symbols each corresponding to a coefficient type;

- means for determining a set of quantizers each associated with a corresponding coefficient type such that an estimated quality parameter in the spatial domain, expressed as a function of estimated distortions each associated with a quantizer of the set, meets a predetermined criterion and such that the sum of rates each associated with a quantizer of the set is minimised;

- means for dequantizing at last one symbol corresponding to a given coefficient type into a dequantized coefficient using the quantizer determined for said given coefficient type; and

- means for transforming a plurality of dequantized coefficient into pixel values in the spatial domain for said block.

Optional features proposed above in connection with the encoding method may also apply to the decoding method, encoding device and decoding device just mentioned.

<9th & 10th aspects>

9th and 10th aspects of the present invention, set out below, may be used for the encoding of any image, regardless of the enhancement nature of the image, and furthermore regardless of its resolution.

These aspects of the invention are particularly advantageous when encoding images without prediction.

The features of the 9th and 10th aspects of the invention may be provided in combination with the features of the 1st to 4th and/or 5th to 8th aspects but this is not essential and it is possible to use the features of the 9th and 10th aspects independently of the features of the 1st to 4th aspects and of the 5th to 8th aspects.

According to the 9th aspect of the present invention, there is provided a method for encoding at least one block of pixels, the method comprising:
- transforming pixel values for said block into a set of coefficients each having a coefficient type;
- determining, for each coefficient type, an estimated value representative of a ratio between a distortion variation provided by encoding a coefficient having the concerned type and a rate increase resulting from encoding said coefficient;
- subjecting coefficients of said set to a quantization step to produce quantized symbols, wherein the subjected coefficients form a subset of said set and wherein the estimated values for coefficient types of coefficients in said subset are larger than the highest estimated value over coefficient types of coefficients not included in said subset; and
- encoding the quantized symbols.

Thanks to the estimated value or ratio for each coefficient type, it is possible to order the various coefficient types by decreasing estimated value, i.e. by decreasing merit of encoding as explained below, and the encoding process may be applied only to coefficients having the higher values (i.e. ratios), forming the subset defined above.

Said distortion variation is for instance provided when no prior encoding has occurred for the coefficient having the concerned type. This amounts to taking into account the initial merit which makes it possible to keep an optimal result, as shown below.

The method may include a step of computing said pixel values by subtracting values obtained by decoding a base layer to values representing pixels of an image. The pixel values are for instance representative of residual data to be encoded into an enhancement layer.

The estimated value for the concerned coefficient type may be computed depending on a coding mode of a corresponding block in the base layer (i.e. for each of a plurality of such coding modes).

The following steps may also be included:
- for each of a plurality of possible subsets each comprising a respective number of first coefficients when coefficients are ordered by decreasing estimated value of their respective coefficient type, selecting quantizers for coefficients of the concerned possible subset such that the distortions associated with the selected quantizers meet a predetermined criterion;
- selecting, among said possible subsets, the subset minimising the rate obtained by using the quantizers selected for said subset, wherein the subset of subjected coefficients is the selected subset.

The number of coefficients to be quantized and encoded is thus determined during the optimisation process. A plurality of possible subsets are considered; however, as the coefficient types are ordered by decreasing encoding merit, only N+1 subsets need be considered if N is the total number of coefficients.

For instance, a quantizer is selected for each of a plurality of coefficient types, for each of a plurality of block sizes and for each of a plurality of base layer coding mode.

The step of selecting quantizers may be performed by selecting, among optimal quantizers each associated with a rate and corresponding distortion, an optimal quantizer for each coefficient type associated with the possible subset concerned, such that the sum of the rates associated with selected quantizers is minimal and the global distortion resulting from the distortions associated with said selected quantizers corresponds to a predetermined distortion. Such an implementation is particularly interesting in practice.

For each coefficient type, an optimal quantizer may be selected for each of a plurality of block sizes and for each of a plurality of base layer coding mode.

The method may further include, for at least one coefficient type, a step of determining a probabilistic model for coefficients of said at least one coefficient type based on a plurality of values of coefficients of said at least one coefficient type, wherein said estimated value for said at least one coefficient type is computed based on said probabilistic model. Ordering of coefficients according to their encoding merit may thus be performed based on the probabilistic model identified for the various coefficient types, which is a convenient way to take into account effective values of the coefficient in the process.

Said estimated value for a given coefficient type may in practice be computed using a derivative of a function associating rate and distortion of optimal quantizers for said coefficient type. Such rate and distortion of optimal quantizers are for example stored for a great number of possible values of the various parameters, as explained below. This allows a practical implementation.

Precisely, said estimated value for a given coefficient type may be determined by computing $\frac{2\sigma^2}{f_n'(0)}$, where $\sigma_n$ is the standard variation among
coefficients of said type and \( f \) is a function associating rate \( R_n \) and distortion \( D_n \) of optimal quantizers for coefficients of said type \( n \) and defined as follows:

\[
R_n = f_n(-\ln(D_n / \sigma_n)).
\]

For instance, \( f_n(0) \) is determined using values stored in association with the determined probabilistic model.

The step of transforming pixel values corresponds for instance to a transformation from the spatial domain (pixels) to the frequency domain (e.g. into coefficients each corresponding to a specific spatial frequency). The transforming step includes for instance applying a block based Discrete Cosine Transform; each of said coefficient types may then correspond to a respective coefficient index.

According to the 10th aspect of the present invention, there is provided apparatus for encoding at least one block of pixels, comprising:
- means for transforming pixel values for said block into a set of coefficients each having a coefficient type;
- means for determining, for each coefficient type, an estimated value representative of a ratio between a distortion variation provided by encoding a coefficient having the concerned type and a rate increase resulting from encoding said coefficient;
- means for subjecting coefficients of said set to a quantization step to produce quantized symbols, wherein the subjected coefficients form a subset of said set and wherein the estimated values for coefficient types of coefficients in said subset are larger than the highest estimated value over coefficient types of coefficients not included in said subset; and
- means for encoding the quantized symbols.

Optional features proposed above in connection with the encoding method may also apply to the encoding apparatus just mentioned.

11th to 14th aspects of the present invention, set out below, may be used for the encoding of any image, regardless of the enhancement nature of the image, and furthermore regardless of its resolution.

These aspects of the invention are particularly advantageous when encoding images without prediction.

The features of the 11th to 14th aspects of the invention may be provided in combination with the features of the 1st to 4th and/or 5th to 8th and/or 9th and 10th aspects but this is not essential and it is possible to use the features of the 11th to 14th aspects...
independently of the features of the 1st to 4th aspects and of the 5th to 8th aspects and of the 9th and 10th aspects.

According to the 11th aspect of the invention, there is provided a method for encoding a first image comprising blocks of pixels, the method comprising:

- transforming blocks of pixel data into blocks of transformed block coefficients;
- quantizing the transformed block coefficients into quantized symbols;
- obtaining alphabets of symbols, each alphabet defining the possible symbols of an associated quantized transformed block coefficient;
- for each alphabet, obtaining a probabilistic distribution of the possible symbols defined therein;
- using the probabilistic distributions to group at least two alphabets into a new alphabet, the new alphabet defining, as symbols, the combinations of possible symbols of the at least two transformed block coefficients associated with the grouped alphabets;
- obtaining binary entropy codes for the possible symbols of each remaining alphabet; and
- encoding the quantized symbols using the obtained binary codes.

This aspect of the invention is applicable to any block-based video codec for which a probabilistic distribution of transformed coefficients, such as DCT coefficients, is available. The probabilistic distribution of the corresponding quantized symbols may then be directly obtained, based on parameters of the quantization (i.e. of the used quantizer).

The grouping of alphabets is to be understood as a product of those alphabets, as set out for example in the book "Elements of Information Theory" (T.M. Cover, J.A. Thomas, Second Edition., ed. Wiley, 2006).

For instance, the alphabet resulting from the grouping of two alphabets \( A = \{a_1, \ldots, a_J\} \) and \( B = \{b_1, \ldots, b_J\} \) is made of all the pairs of symbols \( (a_i, b_j) \).

Thanks to this aspect of the invention, the mean length (i.e. entropy) of the used binary codes is substantially improved compared to the conventional approaches, and is made closer to the theoretical entropy of the probabilistic distributions. Better compression of the image is thus obtained.

This is due to the grouping of alphabets as recited above, which corresponds to using only one symbol (one binary code) to encode together at least two quantized coefficients. Indeed, the grouping of alphabets makes it possible to drastically reduce the maximum overhead for each symbol to encode, which is one bit
for the conventional Huffman coding. For example, by grouping $N$ same alphabets, the maximum overhead for one symbol to encode falls to $1/N$ bit on average for each grouped alphabet.

Furthermore, according to this aspect of the invention, the grouping of alphabets is based on the probabilistic distributions of the symbols, enabling efficient grouping of the alphabets to improve the mean length of the binary codes used. This is because, as will become apparent below, the probabilistic distributions make it possible to distinguish the alphabets having small entropy and those having high entropy, to adapt the grouping. An alphabet with small entropy (i.e. generally having the worst mean overhead) may then be "hidden" into a new alphabet with an alphabet having high entropy (which has a low mean overhead).

According to the 12th aspect of the present invention, there is provided apparatus for encoding a first image comprising blocks of pixels, the apparatus comprising:

- a transformation module for transforming blocks of pixel data into blocks of transformed block coefficients;
- a quantization filter for quantizing the transformed block coefficients into quantized symbols;
- an alphabet generator for generating alphabets of symbols, each alphabet defining the possible symbols of an associated transformed block coefficient due to the quantization;
- a probabilistic module for obtaining, for each alphabet, a probabilistic distribution of the possible symbols defined therein;
- a grouping module configured to group, using the probabilistic distributions, at least two alphabets into a new alphabet, the new alphabet defining, as symbols, the combinations of possible symbols of the at least two transformed block coefficients associated with the grouped alphabets;
- a binary code generator for generating, for each remaining alphabet, binary entropy codes for the possible symbols of each remaining alphabet; and
- an encoding module for encoding the quantized symbols using the obtained binary codes.

According to the 13th aspect of the present invention, there is provided a method for decoding a data bit-stream of an encoded first image, the method comprising:
- obtaining, from the bit-stream, blocks of encoded symbols corresponding to quantized block coefficients;
- obtaining alphabets of symbols, each alphabet defining the possible symbols of an associated quantized block coefficient;

5 - for each alphabet, obtaining a probabilistic distribution of the possible symbols defined therein;
- using the probabilistic distributions to group at least two alphabets into a new alphabet, the new alphabet defining, as symbols, the combinations of possible symbols of the at least two quantized block coefficients associated with the grouped alphabets;
- obtaining binary entropy codes for the possible symbols of each remaining alphabet;
- decoding the encoded symbols using the obtained binary codes.

According to the 14th aspect of the present invention, there is provided apparatus for decoding a data bit-stream of an encoded first image, the apparatus comprising:
- a parser for obtaining, from the bit-stream, blocks of encoded symbols corresponding to quantized block coefficients;
- an alphabet generator for generating alphabets of symbols, each alphabet defining the possible symbols of an associated quantized block coefficient;

10 - a probabilistic module for obtaining, for each alphabet, a probabilistic distribution of the possible symbols defined therein;
- a grouping module configured to group, using the probabilistic distributions, at least two alphabets into a new alphabet, the new alphabet defining, as symbols, the combinations of possible symbols of the at least two quantized block coefficients associated with the grouped alphabets;
- a binary code generator for generating binary entropy codes for the possible symbols of each remaining alphabet; and
- a decoding module for decoding the encoded symbols using the obtained binary codes.

The encoding/decoding apparatuses and the decoding method may have features and advantages that are analogous to those set out above and below in relation to the encoding method embodying the 11th aspect, in particular that of improving the mean length of the binary codes used.
In particular, the encoding method may further comprise determining an entropy value for each of the alphabets, and the grouping of at least two alphabets comprises grouping the alphabet having the smallest entropy with the alphabet having the highest entropy.

The grouping can then be achieved with few computations, since the entropy values can be determined directly from the probabilistic distribution of the corresponding symbols.

Here, the basic idea is that the alphabets with the largest overhead are most of the time the alphabets with the smallest entropy.

This is because the coding overhead for conventional encoding (i.e. without grouping of alphabets) is mainly due to symbols with high probability of occurrence. Clearly, one understands that the probabilities close to one are in alphabets with small entropy.

On the other hand, alphabets with high entropy cannot have high probabilities.

Finally, in the product alphabet (i.e. after grouping), the largest probability is bounded by the largest probability of the alphabet with high entropy. So, it is ensured that no high probability remains after grouping and the coding overhead is thus reduced.

In a variant, the grouping of at least two alphabets comprises grouping two alphabets which maximizes a reduction in distance between the entropy of the alphabets and the entropy of the binary codes constructed from the alphabets.

The entropy of the binary codes may also be seen as the effective mean length of those codes.

The reduction may be understood as a difference between the above-defined distance when using both alphabets before grouping, and the above-defined distance when using the resulting new alphabet.

This approach is more exhaustive since the decision to group alphabets is directly taken from the overhead due to not grouping those alphabets.

According to a first particular feature, each distance relating to an alphabet is weighted by the entropy of that alphabet.

This strategy gives priority to grouping alphabets having small entropy rather than alphabets with larger entropy, when those alphabets have binary codes with similar distances to the entropy.
According to a second particular feature, the entropy of the binary codes constructed from the alphabets is modelled by the entropy of a corresponding Shannon code.

Usually the binary codes are generated using Huffman code. However the entropy (or mean length) of Huffman codes cannot be anticipated without constructing the Huffman tree itself, which is a very costly and not practical operation, in particular when an iterative grouping process is implemented.

Using the entropy of Shannon codes to model that entropy of the binary codes used is therefore a low-cost approach, which furthermore provides an efficient determination of the reductions in distance. This is because a distance of the Shannon code to the entropy provides an upper bound of the corresponding Huffman code distance to the entropy.

In one embodiment of the invention, the method further comprises iteratively grouping alphabets using the probabilistic distributions, wherein the probabilistic distribution of a new alphabet is calculated based on the probabilistic distributions of the at least two alphabets involved in the corresponding grouping.

This iterative process makes it possible to group alphabets several times, thus further reducing the mean coding overhead per symbol. The encoding of the first image is therefore made more efficient in terms of bitrate.

According to a particular feature, alphabets are iteratively grouped as long as a resulting new alphabet comprises less symbols than a predefined number.

Such limit of the number of symbols is defined to ensure that the associated Huffman trees (or equivalent) are small enough to be handled in real time by the codec. Indeed, an alphabet with a very large number of symbols will require a lot of memory to generate a corresponding Huffman tree.

In another embodiment of the invention, the step of grouping groups at least two alphabets associated with at least two transformed block coefficients of the same block of pixels.

This strategy defines an intra-block grouping.

Grouping within a block maintains the random spatial accessibility of the encoding stream with a precision of one block, in particular when temporal and spatial predictions are disabled.

In particular, the same grouping of alphabets is performed for a plurality of blocks within the first image. This plurality may comprise all the blocks of the first image, or a set of blocks that are collocated with base layer blocks encoded with the
same coding mode (e.g. Intra prediction, P image Inter prediction, B image Inter prediction, Skip mode). Indeed, such blocks often use the same quantizations, and then the same alphabets for their symbols to encode.

A result of this is that with few calculations entropy encoding is improved for a substantial part of the image.

According to a particular feature, the at least two transformed block coefficients are adjacent coefficients within the same block.

According to a particular embodiment, the encoding method further comprises grouping at least two alphabets associated with transformed block coefficients collocated in at least two blocks of pixels.

Of course, those grouped alphabets may be alphabets already grouped by the intra-block grouping (i.e. representing several block coefficients with a block).

This strategy defines an inter-block grouping.

It takes advantage of some Markov-like correlation between neighbouring or adjacent blocks to efficiently reduce the mean overhead per symbol.

In particular, the at least two blocks of pixels are from the same macroblock dividing the first image.

This maintains the spatial random accessibility of the encoding stream with a precision of a macroblock, in particular when temporal and spatial predictions are avoided.

Furthermore, the non-dependency from macroblock to macroblock also enables massive parallelization of the encoding and decoding processes, as well as low memory bandwidth consumption, since all encoding/decoding operations can be performed locally in each macroblock, independently from the other macroblocks.

This results in minimizing the amount of data transfer in memory, which is crucial for a low-complexity video codec design.

According to a particular feature, a new alphabet resulting from the grouping of at least two alphabets replaces the at least two grouped alphabets.

According to another particular feature, the probabilistic distribution is the same for the alphabets associated with transformed block coefficients collocated within a plurality of the blocks of the first image. The plurality may comprise all the blocks of the first image, but also a part of it (e.g. related to the same base coding mode).

This reduces the amount of information to handle.

In particular, the obtaining of the probabilistic distribution for an alphabet associated with a transformed block coefficient comprises fitting a Generalized
Gaussian Distribution (GGD) model onto the transformed block coefficients collocated with said transformed block coefficient within said plurality of blocks. The probabilistic distribution for the alphabet can then be directly deduced from the fitted GGD, using the quantization parameters.

According to an embodiment of the invention, the encoding method further comprises, prior to grouping, discarding alphabets having associated entropy or a size (in terms of number of symbols) less than a predefined value.

This is because the impact of these alphabets on the PSNR (Peak signal-to-noise ratio) is negligible. In that case, it is not worth grouping those alphabets.

According to another feature of the invention, the obtaining of binary codes comprises obtaining a Huffman code from each of the remaining alphabets and from their associated probabilistic distributions.

Applying the Huffman coding to the grouped alphabets provides the best entropy encoding for those alphabets.

In one embodiment of the invention, the encoding method further comprises encoding a low resolution version of the first image into an encoded low resolution image; generating a residual enhancement image by subtracting an upsampled decoded version of the encoded low resolution image from said first image; and the step of transforming is applied to the blocks of pixels forming the residual enhancement image.

In other words, the main steps of the encoding method (i.e. the transforming of blocks, the quantizing of coefficients, the obtaining of alphabets, the obtaining of probabilistic distributions, the grouping of alphabets, the obtaining of binary codes and the encoding of the symbols) are applied to the residual enhancement image.

In this particular situation, the invention drastically reduces the complexity of encoding that residual enhancement image compared to conventional encoding schemes, while keeping a good compression ratio if the temporal and spatial predictions are disabled for that encoding.

In particular, the alphabet associated with a transformed block coefficient of a block depends on a coding mode of the block collocated with that block in the encoded low resolution image.

Taking into account the coding mode of the corresponding blocks in the base layer enables fitting a (GGD) distribution to the effective values taken by various transformed block coefficients (DCT coefficients) with more accuracy. Therefore, as the
probabilities of symbols are closer to reality, the entropy coding thus gets closer to the theoretical entropy of the data source to code.

According to a particular feature, the same grouping of alphabets is performed for the blocks within the residual enhancement image that are collocated with blocks encoded with the same encoding mode in the encoded low resolution image.

This reduces the amount of calculation for several blocks while providing an efficient improvement of the entropy coding, since the same alphabet is then used to encode blocks corresponding to the same base coding mode.

The above optional features may also apply optionally to the decoding method.

Furthermore, according to an embodiment of the invention, the decoding method may further comprise, for each alphabet of symbol, obtaining parameters from the bit-stream and applying these parameters to a probabilistic distribution model to obtain the probabilistic distribution of the possible symbols defined in said alphabet. By transmitting only parameters, bitrate of the bit-stream can be saved.

<15th to 19th aspects>

15th to 19th aspects of the present invention, set out below, may be used for the encoding of any image, regardless of the enhancement nature of the image, and furthermore regardless of its resolution.

These aspects of the invention are particularly advantageous when encoding images without prediction.

The features of the 15th to 19th aspects of the invention may be provided in combination with the features of the 1st to 4th and/or 5th to 8th and/or 9th and 10th and/or 11th to 14th aspects but this is not essential and it is possible to use the features of the 15th to 19th aspects independently of the features of the 1st to 4th aspects and of the 5th to 8th aspects and of the 9th and 10th aspects and of the 11th to 14th aspects.

These aspects of the present invention seek to improve entropy encoding, in particular with the aim of getting closer to the entropy rate.

According to the 15th aspect of the present invention, there is provided a method for encoding a first image comprising blocks of pixels, the method comprising:
- obtaining blocks of quantized coefficients corresponding to blocks of the first image;
- obtaining alphabets of values, each defining the possible values of an associated quantized coefficient;
- associating a respective flag with each block of quantized coefficients, said flag specifying a magnitude;
- restricting alphabets corresponding to the quantized coefficients of each block, based on the magnitude specified in the flag associated with that block;
- generating binary entropy codes for the possible values of each restricted alphabet; and
- encoding the quantized coefficients using the generated binary codes.

Thanks to this aspect of the invention, the mean length (i.e. entropy) of the binary codes used is substantially improved compared to the conventional approaches, and is made closer to the theoretical entropy of the probabilistic distributions. Better compression of the image is thus obtained.

This is due to the use of a block flag as defined above, since the magnitude specified therein makes it possible to restrict the alphabets of values (or "symbols") used to encode those quantized coefficients.

Thanks to this restriction of the alphabets, the number of values therein is reduced, also reducing the mean length of the corresponding binary codes. In other words, the bits that would be required for enabling the encoding of unused values (for example those above a maximum magnitude of the block) are avoided with the present aspect of the invention.

As will become apparent below, the overhead resulting from the encoding of the flag may be substantially limited in order to obtain, thanks to the present aspect of the invention, an overall gain in compression of about 5%.

One may note that a flag specifying a particular magnitude (for example equal to zero) may have the same function as the Skip flag=1 of the prior art. In such a case, the encoding of the corresponding block may be avoided. The flag according to the present aspect of the invention is furthermore used to restrict the alphabets in order to further decrease the entropy of the generated binary codes.

According to the 16th aspect of the present invention, there is provided apparatus for encoding a first image comprising blocks of pixels, the encoding apparatus comprising:
- a processing module for generating blocks of quantized coefficients corresponding to blocks of the first image;
- an alphabet generator for generating alphabets of values, each defining the possible values of an associated quantized coefficient;
- a flag generator for associating a respective flag with each block of quantized coefficients, said flag specifying a magnitude;
- an alphabet restriction means for restricting alphabets corresponding to the quantized coefficients of each block, based on the magnitude specified in the flag associated with that block;
- a binary code generator for generating binary entropy codes for the possible values of each restricted alphabet; and
- an encoding module for encoding the quantized coefficients using the generated binary codes.

According to the 17th aspect of the present invention, there is provided a method for decoding a data bit-stream of an encoded first image, the method comprising:
- obtaining, from the bit-stream, blocks of encoded quantized coefficients;
- obtaining alphabets of values for the encoded quantized coefficients, each alphabet defining the possible values taken by a quantizer associated with an encoded quantized coefficient;
- obtaining, from the bit-stream, a flag associated with each block of encoded quantized coefficients, said flag specifying a magnitude;
- restricting alphabets corresponding to the encoded quantized coefficients of each block, based on the magnitude specified in the flag associated with that block;
- generating binary entropy codes for the possible values of each restricted alphabet;
- decoding the encoded quantized coefficients using the generated binary codes.

According to the 18th aspect of the present invention, there is provided apparatus for decoding a data bit-stream of an encoded first image, the decoding apparatus comprising:
- means for obtaining, from the bit-stream, blocks of encoded quantized coefficients;
- an alphabet generator for generating alphabets of values for the encoded quantized coefficients, each alphabet defining the possible values taken by a quantizer associated with an encoded quantized coefficient;
- means for obtaining, from the bit-stream, a flag associated with each block of encoded quantized coefficients, said flag specifying a magnitude;
- an alphabet restriction means for restricting alphabets corresponding to the encoded quantized coefficients of each block, based on the magnitude specified in the flag associated with that block;
- a binary code generator for generating binary entropy codes for the possible values of each restricted alphabet; and
- a decoding module for decoding the encoded quantized coefficients using the generated binary codes.

According to the 19th aspect of the present invention, there is provided a bit-stream of an encoded first image, comprising:
- blocks of binary codes corresponding to blocks of encoded quantized coefficients;
- data (for example parameters of a probabilistic distribution model of the quantized coefficients as described below in more detail) used to determine quantizers associated with the encoded quantized coefficients, each quantizer defining an alphabet of possible values;
- binary flags, each associated with a block of binary codes and specifying a magnitude;

wherein the binary codes of a block belong to binary entropy codes generated from the alphabets restricted based on the magnitude specified in the flag associated with that block.

The bitstream may be carried by a carrier medium such as a signal. It may also be recorded on a recording medium.

The encoding/decoding apparatuses and the decoding method may have features and advantages that are analogous to those set out above and below in relation to the encoding method embodying the 15th aspect of the present invention, in particular that of improving the mean length of the binary codes used.

In particular regarding the encoding method, each flag is selected from a set (or alphabet) of flags, each flag of the set having an associated probability of occurrence.

This makes it possible for the flag associated with each block to be entropy encoded based on the probabilities of occurrence of the flags of the set.

Thanks to these provisions, the overhead due to the flags is limited. This provides an overall improvement thanks to the present aspect of the invention.

According to a particular feature, the set of flags is a restricted set from a larger set of flags. This provision is useful since the number of possible values for the quantized coefficients is very large, thus defining a huge number of possible magnitudes (and therefore flags) for the blocks. Reducing this number to a restricted number enables the flag encoding overhead to be kept as low as possible compared to the compression gain obtained by the restriction of the alphabets as set out above.
According to a feature of this aspect of the invention, the encoding method may further comprise restricting the larger set of flags based on probabilities of occurrence of the flags of that larger set, a probability of occurrence of a flag corresponding to the probability that a block of quantized coefficients has a maximum magnitude equal to the magnitude specified in the flag.

This makes it possible to keep, in the restricted set, the most significant flag values with respect to the probabilities of occurrence.

In particular, restricting the larger set of flags comprises selecting flags of the larger set that define intervals of magnitudes corresponding to a probability of occurrence greater than a predefined amount, for example 10 %. The probability of occurrence of an interval corresponds to the probability that a block of quantized coefficients has a maximum magnitude comprised within that interval. It may be determined from the probabilities of occurrences of the flags of the larger set.

According to a particular feature, the magnitude specified in a flag associated with a block is determined based on the quantized coefficient values of that block. This ensures restricting the alphabets taking into account how they are used within the considered block.

In particular, the magnitude specified in a flag associated with a block represents a maximum magnitude of the quantized coefficient values of that block. This makes it possible to restrict the alphabets in a way that deletes the values above the maximum magnitude (i.e. unused values). The specified maximum magnitude can be taken by one of the quantized coefficient values, but can also be an upper bound as suggested below.

According to a particular feature, the larger set comprises at least flags specifying, as magnitude, all the possible values defined in the alphabets. It is then easy to select from this set a flag corresponding to the maximum magnitude of a block.

In that case of maximum magnitude, restricting the alphabets of a block may comprise deleting, from the alphabets, the possible values that have a magnitude (i.e. absolute value) larger than the magnitude specified in the flag associated with the block.

In a variant to the maximum magnitude, the magnitude specified in a flag associated with a block represents the value $M_B \cdot d_B$, where $M_B$ is the maximum magnitude of the possible values of all the alphabets and $d_B$ is a block distance to the
maximum magnitude defined by
\[ d_B = \min_{i; a_m \neq 0} \left| M_i - \frac{a_m}{16} \right| \]
with \( a_m \) representing the quantized coefficient values of the block, \( A_i \) representing one of the alphabets, \( M_i \) being the maximum magnitude of the possible values of the alphabet \( A_i \), and \( d_B = M_B \) when all the quantized coefficient values \( a_m \) of the block are zero. This provision ensures that the flag used still takes the value zero for a skipped block.

In another variant, the magnitude specified in a flag associated with a block represents the value \( d_B \), where \( d_B \) is defined by
\[ d_B = \min_{i; a_m \neq 0} M - |a_m| \]
representing the quantized coefficient values of the block, \( A_i \) representing one of the alphabets, \( M_i \) being the maximum magnitude of the possible values of the alphabet \( A_i \), and \( d_B = M_B \) when all the quantized coefficient values \( a_m \) of the block are zero. This value \( d_B \) defines the minimum amount of possible values that can be deleted from each edge (or extreme) of the intervals of possible values of the alphabets, while keeping a lossless entropy encoding of the quantized coefficient values \( a_m \). This is because, due to the above construction of \( d_B \), it is sure that the \( d_B \) most leftward and \( d_B \) most rightward values of those intervals (i.e. the \( d_B \) most positive and negative values) are not used in the considered block.

In those cases, restricting the alphabets of a block may comprise deleting the \( d_B \) most negative and \( d_B \) most positive possible values from each of the alphabets.

According to a particular feature, the number of flags in the restricted set is at most three. It appears that the best results are obtained in that situation. As a variant four flags could however be kept.

In particular, one flag in the restricted set is the flag specifying the largest magnitude in the larger set. This ensures that every block can be entropy encoded without loss, because, when this flag is used for a block, the alphabets are not restricted at all when encoding that block.

In a variant or in combination, one flag in the restricted set specifies a magnitude equal to zero. In particular when the flag magnitude represents the maximum magnitude of all the quantized coefficients in the block, this specific flag has the same function as the skip flag=1 defined in H.264, meaning that all the quantized coefficients of the block are equal to zero. This provision is thus compliant with the Skip
mode of H.264. Furthermore, using such flag has the advantage that there is no need to encode the corresponding blocks, saving bits.

In one embodiment of the invention, the flags associated with the blocks of the same macroblock are concatenated into one flag associated with the macroblock. This ensures a more efficient encoding of that information, based on the grouping of alphabets as described below with more detail.

In that case, the flag associated with the macroblock may be entropy encoded based on probabilities of occurrence of the possible values for such a macroblock flag, those possible values resulting from the product of the sets of flags from which the concatenated flags are selected.

This is because the grouping (or product) of alphabets (or sets) when encoding shares the maximum one-bit Huffman coding overhead between the grouped alphabets. For example, by grouping N same alphabets, the maximum overhead for one value to encode falls to 1/N bit on average for each grouped alphabet.

In particular, the encoding method further comprises obtaining at least one probability \( p_{MB,n} \) (in particular \( p_{MB,0} \)) of occurrence that a macroblock comprises only quantized coefficients having a magnitude less than a given corresponding value \( n \) (i.e. a macroblock with only coefficients equal to zero when considering \( p_{MB,0} \)); and the probabilities of occurrence of the possible values for a macroblock flag depend on said at least one probability \( p_{MB,n} \).

This is because, due to the content dependence between neighbouring blocks within an image, the probability \( p_{MB,0} \) of a skipped macroblock (i.e. when the corresponding four flags have a maximum magnitude equal to 0) is much higher than the product of the four probabilities \( p_{BO} \) of having each block of the macroblock skipped: \( p_{MB,0} > (p_{BO})^4 \). The same applies when considering other maximum magnitude: \( p_{MB,i^2} > (p_{BO})^4 \). The above provision then takes into account this situation to normalize the probabilities of each macroblock flag. The entropy encoding of the macroblock flags based on those probabilities is therefore closer to the true entropy.

According to a particular feature, the sets of flags from which the concatenated flags are selected are the same set restricted from a larger set of flags. Only few probabilities associated with the flags of that restricted set have to be handled in that case.

In another embodiment of the present aspect of the invention, the encoding method further comprises encoding a low resolution version of the first image into an encoded low resolution image; generating a residual enhancement image by
subtracting an upsampled decoded version of the encoded low resolution image from said first image; and wherein the blocks of quantized coefficients are obtained from the residual enhancement image.

In other words, this means that the encoding method of the present aspect of the invention based on restricting the alphabets applies to the residual enhancement image.

In this particular situation, the present aspect of the invention reduces the complexity of encoding that residual enhancement image compared to conventional encoding schemes, while keeping a good compression ratio if the temporal and spatial predictions are disabled for that encoding.

In particular, the flag associated with a block depends on a coding mode of the block collocated with that block in the encoded low resolution image.

Taking into account the coding mode of the corresponding blocks in the base layer enables fitting a (GGD) distribution to the effective values taken by various transformed block coefficients (DCT coefficients) with more accuracy. Therefore, as the probabilities of symbols are closer to reality, the entropy coding thus gets closer to the theoretical entropy of the data source to code.

Optional features are also provided for the decoding method.

For example, the decoding method may further comprise obtaining, from the bit-stream, probabilities associated with each of a set of possible flags; and entropy decoding encoded flags from the bit-stream based on the obtained probabilities to obtain said flag associated with each block.

In one embodiment in which the flag represents a maximum magnitude of the quantized coefficients in the associated block, restricting the alphabets of a block may comprise deleting, from the alphabets, the possible values that have a magnitude (i.e. absolute value) larger than the magnitude specified in the flag associated with the block.

In another embodiment in which the flag may convey information $d_B$ about block distance to the maximum magnitude (i.e. $d_B$ depends on the magnitude specified in the flag associated with the block), restricting the alphabets of a block may comprise deleting the $d_B$ most negative and $d_B$ most positive possible values from each of the alphabets.

According to a feature of the present aspect of the invention, the number of flags in the set is restricted to three at most.
According to a particular feature, one flag in the restricted set specifies a magnitude equal to zero.

In one embodiment, the decoding method further comprises obtaining an encoded macroblock flag associated with a macroblock made of blocks; entropy decoding said encoded macroblock flag based on the obtained probabilities; and splitting the decoded macroblock flag into flags associated with the blocks of the macroblock. As mentioned above, using such macroblock flags concatenating the flags for the blocks of the macroblock substantially improves the entropy encoding of the flags.

In another embodiment of the present aspect of the invention, the decoding method further comprises decoding a first bit-stream to obtain a decoded low resolution version of the first image; obtaining a decoded residual enhancement image from the decoded quantized coefficients; and adding the obtained decoded residual enhancement image to an upsampled version of the decoded low resolution version of the first image, to obtain a decoded first image.

<20th & 21st aspects>

In the absence of temporal prediction in the enhancement layer, decoding a first enhancement image may involve using the motion information of the corresponding first base image in order to obtain residual blocks from a "reference" decoded UHD image (temporally corresponding to the decoded reference base image used for predicting the first base image). Such blocks may then be used to correct the enhancement image data directly obtained from the bit-stream.

Using temporal prediction information of the base layer to decode the enhancement layer is known from the standard SVC (standing for "Scalable Video Coding"). One may note that such approach cannot be applied to blocks which are not temporally predicted in the base layer (i.e. the so-called Intra images or the intra-predicted blocks).

However, the 20th and 21st aspects of the invention intend to improve the efficiency of a decoding method based on predicting the enhancement layer. This aims at improving the quality of reconstructed high resolution (e.g. UHD) images, while keeping low complexity at the encoding and decoding sides.

The features of the 20th and 21st aspects of the invention may be provided in combination with the features of the 1st to 4th and/or 5th to 8th and/or 9th and 10th and/or 11th to 14th and/or 15th to 19th aspects but this is not essential and it is possible to use the features of the 20th and 21st aspects independently of the features of the 1st
to 4th aspects and of the 5th to 8th aspects and of the 9th and 10th aspects and of the 11th to 14th aspects and of the 15th to 19th aspects.

According to the 20th aspect of the present invention, there is provided a method for decoding a scalable video bit-stream, comprising decoding a base layer from the bit-stream, decoding an enhancement layer from the bit-stream and adding the enhancement layer to the base layer to obtain a decoded video of high resolution images, wherein decoding the enhancement layer comprises:

- decoding, from the bit-stream, and dequantizing encoded transformed coefficients of the enhancement layer,
- using motion information of the base layer to predict residual blocks of coefficients of the enhancement layer, and transforming the predicted residual blocks of coefficients into transformed residual blocks;
- obtaining at least one first probabilistic distribution of the transformed coefficients;
- obtaining at least one second probabilistic distribution of the differences between the dequantized transformed coefficients and the coefficients of the transformed residual blocks; and
- merging the dequantized transformed coefficients and the coefficients of the transformed residual blocks, based on the obtained probabilistic distributions.

In more detail, the method for decoding a scalable video bit-stream, comprises:

- decoding a low resolution version of the video, the decoding of the low resolution version comprising using motion information to temporally predict blocks of a low resolution image from blocks of a decoded reference low resolution image;
- decoding an enhancement version of the video, each enhancement image of the enhancement version having a high resolution and temporally corresponding to a low resolution image of the low resolution video; and
- adding each decoded enhancement image to an up-sampled version of the corresponding low resolution image, to obtain a decoded video of decoded high resolution images;

wherein decoding a first enhancement image temporally corresponding to a first low resolution image comprises:

- decoding, from the bit-stream, and dequantizing blocks of encoded quantized transformed coefficients of the first enhancement image;
- obtaining at least one first probabilistic distribution of the quantized transformed coefficients;
- using the motion information to obtain residual blocks from a decoded reference high resolution image temporally corresponding to the decoded reference low resolution image; and transforming said residual blocks into transformed residual blocks;
- obtaining at least one second probabilistic distribution of the differences between the coefficients of the transformed residual blocks and the dequantized transformed coefficients; and
- merging the dequantized blocks of dequantized transformed coefficients with the transformed residual blocks, based on the first and second probabilistic distributions.

The blocks of the encoded enhancement layer obtained from the bit-stream and the residual blocks obtained using the motion information of the base layer are merged together to form parts of the decoded enhancement image. As explained above this decoded enhancement image is then added to an up-sampled decoded base image to obtain a decoded high resolution (e.g. UHD) image.

This approach refines the quality of the decoded transformed (i.e. DCT) coefficients in the decoder.

According to the present aspect of the invention, the quality of the decoded high resolution image is improved compared to known techniques. This is due to the use of two probabilistic distributions that model both the original transformed coefficients and an error of temporal prediction, when merging the transformed coefficients (e.g. DCT coefficients).

The first probabilistic distribution corresponding to the transformed coefficients encoded in the bit-stream may be obtained from the bit-stream itself, for instance from parameters contained therein. These may represent statistical modelling of the original transformed coefficients (i.e. before quantization and encoding).

The second probabilistic distributions that correspond to the blocks predicted using the motion information of the base layer, provide information about the noise of temporal prediction. In particular they provide modelled information on the difference between those predicted coefficients and the transformed coefficients. Since the original transformed coefficients are not known by the decoder, the decoded and dequantized transformed coefficients known at the decoding side are used in place of the original transformed coefficients. The inventors have observed that using those coefficients rather than the original ones provides modelling that is quite close to reality.
Since the decoded transformed coefficients and the transformed predicted coefficients (or residual blocks) both bring relevant information about the original transformed coefficients (DCT coefficients before encoding), using the above probabilistic distributions enables the obtaining of transformed coefficients to be statistically optimized so as to be closer to the original values than in the known techniques.

In particular, probabilities, such as the expected value example below, provide good statistical results.

For example, for low bitrate (meaning large quantization intervals), the temporally predicted blocks may more often provide relevant information on the original DCT coefficients than the quantization level obtained by the dequantization. For high bitrate, the opposite occurs.

The invention allows gains up to several dBs in rate-distortion performance at almost no cost of additional complexity at the decoder, and at the cost of zero additional rate when the parameters have already been transmitted.

As a further advantage, the approach according to the present aspect of the invention does not necessarily have to be performed at the decoding side. For example, it may be switched off in case of very low complexity decoders. Further, the encoding is independent of the switching decision.

According to the 21st aspect of the present invention, there is provided apparatus for decoding a scalable video bit-stream, comprising a base layer decoder configured to decode a base layer from the bit-stream, an enhancement layer decoder configured to decode an enhancement layer from the bit-stream and a video building unit configured to add the enhancement layer to the base layer to obtain a decoded video, wherein the enhancement layer decoder is further configured to:

- decode, from the bit-stream, and dequantize encoded transformed coefficients of the enhancement layer;
- use motion information of the base layer to predict residual blocks of coefficients of the enhancement layer, and transform the predicted residual blocks of coefficients into transformed residual blocks;
- obtain at least one first probabilistic distribution of the transformed coefficients;
- obtain at least one second probabilistic distribution of the differences between the dequantized transformed coefficients and the coefficients of the transformed residual blocks; and
merge the dequantized transformed coefficients and the coefficients of the transformed residual blocks, based on the obtained probabilistic distributions.

In more detail, the decoding apparatus comprises:

- a base decoder configured to decode a low resolution version of the video, using motion information to temporally predict blocks of a low resolution image from blocks of a decoded reference low resolution image;

- an enhancement decoder configured to decode an enhancement version of the video, each enhancement image of the enhancement version having a high resolution and temporally corresponding to a low resolution image of the low resolution video; and

- an image building unit configured to add each decoded enhancement image to an up-sampled version of the corresponding low resolution image, to obtain a decoded video of decoded high resolution images;

wherein the enhancement decoder is further configured to:

- decode, from the bit-stream, and dequantize blocks of encoded quantized transformed coefficients of the first enhancement image;

- obtain at least one probabilistic distribution of the quantized transformed coefficients;

- use the motion information to obtain residual blocks from a decoded reference high resolution image temporally corresponding to the decoded reference low resolution image; and transform said residual blocks into transformed residual blocks;

- obtain at least one second probabilistic distribution of the differences between the coefficients of the transformed residual blocks and the dequantized transformed coefficients; and

- merge the dequantized blocks of dequantized transformed coefficients with the transformed residual blocks, based on the first and second probabilistic distributions.

For example, the step of merging may merge a dequantized transformed coefficient with a collocated coefficient in the transformed residual blocks (meaning collocated blocks and collocated coefficients within those blocks), using first and second probabilistic distributions associated with these collocated coefficients, on a quantization interval associated with the value of the corresponding quantized transformed coefficient (i.e. the value before the quantized transformed coefficient is dequantized).
This ensures an accurate merged transformed coefficient to be provided, given its quantized value that has been transmitted in the encoded bit-stream.

In particular, the first and second probabilistic distributions are integrated using Riemann sums over that quantization interval during the merging step. This provision makes it possible to perform a probabilistic merger of transformed coefficients, on low complexity decoders.

According to a particular feature, the step of merging comprises calculating the expectation of a block coefficient, given the quantization interval associated with the value of the corresponding quantized transformed coefficient and given its corresponding value in the transformed residual blocks, based on the first and second probabilistic distributions.

In particular, calculating the expectation \( \tilde{X}_i \) of a block coefficient \( i \) comprises calculating the following value:

\[
\tilde{X}_i = \frac{\int_{-\infty}^{\infty} x \cdot \text{PDF}_i(x) \cdot \text{PDF}_N(x - Y_0) \, dx}{\int_{-\infty}^{\infty} \text{PDF}_i(x) \cdot \text{PDF}_N(x - Y_0) \, dx}
\]

where \( \text{PDF}_i \) is the first probabilistic distribution associated with the block coefficient \( i \), \( \text{PDF}_N \) is the second probabilistic distribution, \( Y_0 \) is the value of the coefficient collocated with said block coefficient \( i \) in the transformed residual blocks, and \( Q_m \) is the quantization interval associated with the value of the quantized transformed coefficient collocated with said block coefficient \( i \).

These approaches combine the probabilities of occurrence of the considered coefficient in the quantization interval (i.e. the first probabilistic distribution), the predicted value (\( Y_0 \)) and the noise modelling of the prediction (i.e. the second probabilistic distribution).

The probabilistic best value is thus obtained for the transformed coefficients when reconstructing the decoded high resolution images. These images prove to be statistically improved with respect to quality.

In one embodiment of the invention, the probabilistic distributions are generalized Gaussian distributions

\[
\text{GGD}(\alpha_i; \beta_i, x) = \frac{\beta_i}{2 \alpha_i \Gamma(1 / \beta_i)} \exp \left( -\frac{|x|^{\beta_i}}{\alpha_i \beta_i} \right),
\]

where \( \alpha_i \) and \( \beta_i \) are two parameters. This parametric model is well-suited for modelling noise, such as the residuals.
In particular, the obtaining of the second probabilistic distribution comprises fitting a Generalized Gaussian Distribution model onto the differences between the coefficients in the transformed residual blocks and the dequantized transformed coefficients. In that case, the second probabilistic distribution is statistically obtained based on the coefficients that are actually handled by the decoder.

According to a particular feature, the obtaining of the first probabilistic distribution comprises obtaining parameters from the bit-stream and applying these parameters to a probabilistic distribution model.

In one particular embodiment of the present aspect of the invention, the low resolution or base image temporally corresponding to a first enhancement image to decode is an image bi-directionally predicted from reference low resolution or base images using motion information in each of the two directions, and the decoding of the first enhancement image comprises obtaining transformed residual blocks for each direction and merging together the transformed residual blocks in both directions with the dequantized blocks of dequantized transformed coefficients.

This applies for example to enhancement images corresponding to B-type images of the base layer.

This approach proves to be more precise than an approach which first determines a single transformed residual block based on prediction in both directions. This is because a motion prediction noise estimation in each direction is separately obtained, improving a probabilistic merger.

Similarly to the case briefly described above, the merging can be based on calculating an expectation.

For example, the step of merging may comprise calculating the merger value $\tilde{x}_i$ of a block coefficient $i$ using the formula:

$$\tilde{x}_i = \frac{\int_{Q_0} x \cdot \text{PDF}_f(x) \cdot \text{PDF}_N(x - Y_0) \cdot \text{PDF}_N(x - Y'_{0}) \, dx}{\int_{Q_0} \text{PDF}_f(x) \cdot \text{PDF}_N(x - Y_0) \cdot \text{PDF}_N(x - Y'_{0}) \, dx}$$

where PDF_f is the first probabilistic distribution associated with the block coefficient $i$, PDF_N and PDF_N' are the second probabilistic distributions for respectively each of the two directions, $Y_0$ and $V_0$ are the value of the coefficient collocated with said block coefficient $i$ in the transformed residual blocks in respectively each of the two directions, and $Q_m$ is the quantization interval associated with the value of the quantized transformed coefficient collocated with said block coefficient $i$. 

In one embodiment of the present aspect of the invention, obtaining residual blocks comprises:
- obtaining, using the motion information, motion predictor blocks from a decoded reference high resolution image;
- up-sampling the low resolution image temporally corresponding to the first enhancement image to decode into high resolution, to obtain up-sampled blocks;
- subtracting each motion predictor block from a corresponding (i.e. collocated) up-sampled block to obtain the residual blocks.

These steps define the temporal prediction of the enhancement layer based on the images already reconstructed. They produce another enhancement layer (since each obtained block is the difference with the base layer) from which a modelling of the temporal prediction noise can be performed.

In one embodiment of the present aspect of the invention, before using the motion information, that motion information is up-sampled (or interpolated) into high resolution. This is because the reference image on which that information is about to be used is of high resolution.

According to a particular feature, the motion information that is up-sampled comprises, for a given block, a motion vector and a temporal residual block; and the obtaining of the motion predictor blocks comprises
- obtaining blocks of the decoded reference high resolution image using the up-sampled motion information, and
- adding the up-sampled temporal residual block to the obtained blocks.

It may further comprise a reference image index identifying said reference low resolution image, when the encoding with temporal prediction uses multiple reference images.

In another embodiment of the present aspect of the invention, the decoding method may further comprise filtering, using a deblocking filter, the obtained decoded high resolution images; wherein parameters (e.g. the filter strength parameter or the quantization-dependent parameter) of the deblocking filter depend on the first and second probabilistic distributions used during the merger.

This makes it possible to locally adjust the post-processing for filtering discontinuities resulting from the modelling according to the invention.

In yet another embodiment, the second probabilistic distributions are obtained for blocks collocated with enhancement image blocks of the corresponding low resolution or base image that are encoded with the same coding mode. The coding
mode of the base (low resolution) layer is for example the INTER mode, which may be further subdivided into an INTER P-prediction mode and an INTER B-prediction mode, or the SKIP mode (as defined in H.264).

In another embodiment, first probabilistic distributions are obtained for respectively each of a plurality of channels, wherein a channel is associated with collocated coefficients having the same block coefficient position in their respective blocks. Furthermore, a channel may be restricted to the blocks collocated with base layer blocks having the same coding mode.

<Other aspects>

The sets of aspects described above (5th to 8th aspects, 9th & 10th aspects, 11th to 14th aspects, 15th to 19th aspects and 20th & 21st aspects) can be applied independently of one another but any two or more sets can be combined. As noted above, too, they can also be applied independently of the 1st to 4th aspects or in combination with the 1st to 4th aspects.

In the 1st to 4th aspects of the present invention the encoding and decoding of the base layer are in conformity with HEVC. However, in other aspects of the present invention, it is possible to encode and decode the base layer using techniques other than HEVC, for example H.264. Other preferred features of the 1st to 4th aspects of the invention, as described above, can still be applied when HEVC encoding and decoding are not used. Combinations with the sets of aspects described above (5th to 8th aspects, 9th & 10th aspects, 11th to 14th aspects, 15th to 19th aspects and 20th & 21st aspects) can be made when HEVC encoding and decoding are not used.

<Program aspects>

Any of the methods and apparatuses embodying the aforesaid 1st to 18th, 20th and 21st aspects of the present invention and the Other Aspects may be implemented in software.

Accordingly, further aspects of the present invention relate to programs which, when executed by a computer or processor, cause the computer or processor to carry out encoding methods embodying any of, or any combination of, the 1st, 5th, 9th, 11th and 15th aspects of the present invention.

Further aspects of the present invention relate to programs which, when executed by a computer or processor, cause the computer or processor to carry out decoding methods embodying any of, or any combination of, the 2nd, 6th, 13th, 17th and 20th aspects of the present invention.
Further aspects of the present invention relate to programs which, when loaded into a computer or processor, cause the computer or processor to function as the encoding apparatuses of any of, or any combination of, the 3rd, 7th, 10th, 12th and 16th aspects of the present invention.

Further aspects of the present invention relate to programs which, when loaded into a computer or processor, cause the computer or processor to function as the decoding apparatuses of any of, or any combination of, the 4th, 8th, 14th, 18th and 21st aspects of the present invention.

The programs may have features and advantages that are analogous to those set out above.

A program embodying the invention may be provided by itself or may be carried in, on or by a carrier medium. The carrier medium may be a storage medium or recording medium. Preferably, the storage medium is computer-readable.

The carrier medium may also be a transmission medium such as a signal. Such a signal may be transmitted through a network such as the Internet to enable a program embodying the present invention to be distributed via the network.

**BRIEF DESCRIPTION OF THE DRAWINGS**

Other particularities and advantages of the invention will also emerge from the following description, illustrated by the accompanying drawings, in which:

- **Figure 1** schematically shows an encoder for a scalable codec;
- **Figure 2** schematically shows the corresponding decoder;
- **Figure 3** schematically illustrates the enhancement video encoding module of the encoder of **Figure 1**;
- **Figure 3a** is a more detailed schematic illustration of the enhancement video encoding module of the encoder of **Figure 1** according to the invention;
- **Figure 4** schematically illustrates the enhancement video decoding module of the encoder of **Figure 2**;
- **Figure 4a** is a more detailed schematic illustration of the enhancement video decoding module of the decoder of **Figure 2** according to the invention;
- **Figure 5** illustrates the performance of current entropy coding using Huffman codes;
- **Figure 6** illustrates a structure of a 4:2:0 macroblock;
- **Figure 7** illustrates an example of a quantizer based on Voronoi cells;
- Figure 8 schematically illustrates the alphabet of symbols and associated probabilities for the quantizer of Figure 7;
- Figure 9 illustrates the video stream structure for the luminance, with indication of the alphabets involved for DCT coefficients;
- Figure 10 shows general steps of the processing of alphabets according to the invention;
  - Figure 11 illustrates the intra-block grouping of Figure 10;
  - Figure 12 illustrates the inter-block grouping of Figure 10;
  - Figure 13 shows steps of an exemplary method for the intra-block grouping of Figure 11;
- Figure 14 shows steps of an exemplary method for the inter-block grouping of Figure 12;
- Figure 15 shows steps of an entropy encoding method according to the invention;
- Figure 16 shows steps of a corresponding decoding method;
- Figure 17 illustrates the performance of grouping alphabets, in comparison to the performances illustrated in Figure 5;
  - Figure 18 illustrates the spatial random access property;
  - Figure 19 illustrates an implementation of entry points to allow spatial random access as illustrated in Figure 18;
- Figure 20 shows a particular hardware configuration of a device able to implement methods according to the invention;
- Figure 21 shows general steps of the restricting operations according to the invention;
- Figure 22 illustrates the restricted alphabets; and
- Figure 23 illustrates the coding structure of the bitstream with macroblock flags according one embodiment of the invention;
- Figure 24 shows the correspondence between data in the spatial domain (pixels) and data in the frequency domain;
- Figure 25 illustrates an exemplary distribution over two quanta;
- Figure 26 shows exemplary rate-distortion curves, each curve corresponding to a specific number of quanta;
- Figure 27 shows the rate-distortion curve obtained by taking the upper envelope of the curves of Figure 26;
- **Figure 28** depicts several rate-distortion curves obtained for various possible parameters of the DCT coefficient distribution;
- **Figure 29** shows the domain where the optimisation is carried out;
- **Figure 30** depicts curves showing the convergence of the optimisation process used to select the quantizers to be used;
- **Figure 31** is a more detailed schematic illustration of the decoder of Figure 2 according to another embodiment of the invention;
- **Figure 32** illustrates the prediction of the enhancement layer in the decoder of Figure 31;
- **Figure 33** illustrates the probabilistic merging in the decoder of Figure 31; and
- **Figure 34** illustrates the performance of the Figure 31 decoder.

**DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION**

**Overview/principle**

As set out in the introduction, given constraints in terms of limited power and memory access bandwidth, an embodiment of the present invention seeks to provide a UHD codec with low complexity based on scalable encoding. It should be noted that, although the description below refers to UHD and HD, HD is merely an example of a first (lower) resolution and UHD is merely an example of a second (higher) resolution. The invention can be applied to any case in which two or more different resolutions of video data are available or can be obtained. Video data having the first (lower) resolution can, for example, be obtained from video data having the second (higher) resolution by downsampling. Alternatively, some video sources, such as video cameras, may deliver simultaneously video data having the first and second resolutions.

Basically, the UHD video is encoded into a base layer and one or more enhancement layers.

The base layer results from the encoding of a low resolution version (version having the first resolution) of the UHD images, in particular having a HD resolution, with a standard existing codec (e.g. H.264 or HEVC - High Efficiency Video Coding). As stated above, the compression efficiency of such a codec relies on spatial and temporal predictions.

Furthermore, the conventional block-based codec, such as H.264, implements a Skip mode as described for example in US application No
2009/0262835. According to the Skip mode, when a macroblock (or block) residual resulting from the prediction is full of zeros (and possibly when the motion vector for the macroblock is zero [i.e. no motion]), a Skip Macroblock flag is set to 1 in the bit stream, instead of coding the motion vectors or the residual. This mainly reduces the number of 5

bits to be coded, and improves compression rate.

Basically, this Skip mode takes advantage of the dependence between the quantized DCT coefficients, in particular of the strong correlation between the small values of the quantized DCT coefficients. For instance, it is generally observed that there are much more DCT blocks in which all the quantized DCT coefficients are zeroes (so-called skipped blocks) than expected from a theory without dependence. In other words, the probability of a skipped block is much higher than the product of probabilities of having a zero for all quantized DCT coefficients.

Further to the encoding of the base layer, an enhancement image is obtained from subtracting an interpolated (or up-scaled or upsampled) decoded image of the base layer from the corresponding original UHD image. The enhancement images, which are residuals or pixel differences with UHD resolution, are then encoded into an enhancement layer.

**Figure 1** illustrates such approach at the encoder 10. An input raw video 11, in particular a UHD video, is down-sampled 12 to obtain a so-called base layer, for example with HD resolution, which is encoded by a standard base video coder 13, for instance H.264/AVC or HEVC. This results in a base layer bit stream 14.

To generate the enhancement layer, the encoded base layer is decoded 15 and up-sampled 16 into the initial resolution (UHD in the example) to obtain the up-sampled decoded base layer.

The latter is then subtracted 17, in the pixel domain, from the original raw video to get the residual enhancement layer X.

The information contained in X is the error or pixel difference due to the base layer encoding and the up-sampling. It is also known as a "residual".

A conventional block division is then applied, for instance a homogenous 8x8 block division (but other divisions with non-constant block size are also possible).

Next, a block-based DCT transform 18 is applied to each block to generate DCT blocks forming the DCT image \( X_{DCT} \) having the initial UHD resolution.

This DCT image \( X_{DCT} \) is encoded in \( X_{DCT}^{ENC} \) by an enhancement video encoding module 19 into an enhancement layer bit stream 20.
The encoded bit-stream EBS resulting from the encoding of the raw video 11 is made of:
- the base layer bit-stream 14 produced by the base video encoder 13;
- the enhancement layer bit-stream 20 encoded by the enhancement video encoder 19; and
- parameters 21 determined and used by the enhancement video encoder.

Examples of those parameters are given here below.

Figure 2 illustrates the associated processing at the decoder 30 receiving the encoded bit-stream EBS.

Part of the processing consists in decoding the base layer bit-stream 14 by the standard base video decoder 31 to produce a decoded base layer. The decoded base layer is then up-sampled 32 into the initial resolution, i.e. UHD resolution.

In another part of the processing, both the enhancement layer bit-stream 20 and the parameters 21 are used by the enhancement video decoding and dequantization module 33 to generate $X_{q^i}^{DE}$. The image $X_{q^i}^{DE}$ is the result of the quantization and then the inverse quantization on the original image X.

An inverse DCT transform 34 is then applied to each block of the dequantized image to obtain the decoded residual $X_{IDCT,q^i}^{DEC}$ (of UHD resolution) in the pixel domain. For example, each decoded residual $X_{IDCT,q^i}^{DEC}$ is added 35 to the corresponding block in the up-sampled decoded base layer to obtain decoded images of the video.

Filter post-processing, for instance with a deblocking filter 36, is finally applied to obtain the decoded video 37 which is output by the decoder 30.

Reducing UHD encoding and decoding complexity relies on simplifying the encoding of the enhancement images at the enhancement video encoding module 19 compared to the conventional encoding scheme.

To that end, the inventors dispense with the temporal prediction and possibly the spatial prediction when encoding the UHD enhancement images. This is because the temporal prediction is very expensive in terms of memory bandwidth consumption, since it often requires accessing other enhancement images as reference images. Low-complexity codecs may then be designed, in particular at the encoding side.

While this simplification reduces by 80% the slow memory random access bandwidth consumption during the encoding process, not using those powerful video
compression tools may deteriorate the compression efficiency, compared to the conventional standards.

In this context, the inventors have developed several additional tools for increasing the efficiency of the encoding of those enhancement images.

**Figure 3** illustrates an embodiment of the enhancement video encoding module 19 (or "enhancement layer encoder") that is provided by the inventors.

In this embodiment the enhancement layer encoder models 190 the statistical distribution of the DCT coefficients within the DCT blocks of a current enhancement image by fitting a parametric probabilistic model.

This fitted model becomes the channel model of DCT coefficients and the fitted parameters are output in the parameter bit-stream 21 coded by the enhancement layer encoder. As will become more clearly apparent below, a channel model may be obtained for each DCT coefficient position within a DCT block based on fitting the parametric probabilistic model onto the corresponding collocated DCT coefficients throughout all the DCT blocks of the image \( X_{DCT} \) or of part of it.

Based on the channel models, quantizers may be chosen 191 from a pool of pre-computed quantizers dedicated to each parametric channel.

The chosen quantizers are used to perform the quantization 192 of the DCT image \( X_{DCT} \) to obtain the quantized DCT image \( X_{DCT,Q} \).

Lastly, an entropy encoder 193 is applied to the quantized DCT image \( X_{DCT,Q} \) to compress data and generate the encoded DCT image \( X_{ENC,DCT,Q} \) which constitutes the enhancement layer bit-stream 20.

The associated enhancement video decoder 33 is shown in Figure 4.

From the received parameters 21, the channel models are reconstructed and quantizers are chosen 330 from the pool of quantizers.

An entropy decoder 331 is applied to the received enhancement layer bit-stream 20 \( (X = X_{ENC,DCT,Q}) \) to obtain the quantized DCT image \( X_{DCT,Q} \).

A dequantization 332 is then performed by using the chosen quantizers, to obtain \( X_{DCT,Q}^{dec} \).

The channel modelling, the selection of quantizers and the adaptation of entropy coding are some of the additional tools as introduced above.
As will become apparent from the explanation below, those additional tools may be used for the encoding of any image, regardless of the enhancement nature of the image, and furthermore regardless of its resolution.

As briefly introduced above, the invention is particularly advantageous when encoding images without prediction.

**Entropy Encoding**

The following section of description focuses on the entropy coding 193. In this respect, predefined quantizers may be used instead of the selection 191 of optimal quantizers.

**Figure** 5 illustrates the performance of conventional entropy encoding.

The y-axis corresponds to the average picture quality obtained when decoding an encoded video bit-stream. The x-axis corresponds to the average bitrate obtained in the encoding of the corresponding video.

The dashed curve shows a rate distortion curve obtained, with bitrate values equal to the theoretical entropy of the coded source. As it is commonly known, the entropy bitrate is the lowest rate bound that can be achieved by any entropy coder.

The plain curve illustrates the rate-distortion curve obtained when a conventional Huffman coding/decoding process is used, for example as defined in the standard H.264.

From these two curves, it can be seen that the bitrate usually obtained is substantially higher than the entropy rate. In particular, this is due to the fact that the Huffman coding may be as much as one bit away from the theoretical entropy for a single symbol to encode.

The present invention seeks to improve this entropy encoding situation, in particular with the aim of getting closer to the entropy rate.

For the detailed description below, focus is made on the encoding and the decoding of a UHD video as introduced above with reference to **Figures 1 to 4**. It is however to be recalled that the invention applies to the encoding of any image from which a probabilistic distribution of transformed block coefficients can be obtained (e.g. statistically). In particular, it applies to the encoding of an image without temporal prediction and possibly without spatial prediction.

Referring again to **Figure 3**, a low resolution version of the initial image has been encoded into an encoded low resolution image, referred above as the base layer; and a residual enhancement image has been obtained by subtracting an interpolated
high resolution (or up-sampled) decoded version of the encoded low resolution image from said initial image.

Conventionally, that residual enhancement image is then transformed from the spatial domain (i.e. pixels) into the (spatial) frequency domain, using for example a block-based DCT transform, to obtain an image of transformed block coefficients. In the Figure, that image is referenced \( X_{DCT} \), which comprises a plurality of DCT blocks, each comprising DCT coefficients.

As an example, the residual enhancement image has been divided into blocks \( B_k \), for instance 8x8 blocks but other divisions may be considered, on which the DCT transform is applied. Within a block, the DCT coefficients are associated with a block coefficient position or "index" \( i \) (e.g. \( i = 1 \) to 64), along a zigzag scan for successive handling when encoding, for example.

Blocks are grouped into macroblocks \( M_k \). A very common case for so-called 4:2:0 YUV video streams is a macroblock made of 4 blocks of luminance \( Y \), 1 block of chrominance \( U \) and 1 block of chrominance \( V \), as illustrated in Figure 6. Here too, other configurations may be considered.

In the example developed below, a macroblock \( M_k \) is made of 16x16 pixels of luminance \( Y \) and the chrominance has been down-sampled by a factor two both horizontally and vertically to obtain 8’8 pixels of chrominance \( U \) and 8’8 pixels of chrominance \( V \). The four blocks within a macroblock \( M_k \) are referenced \( B^1_k, B^2_k, B^3_k, B^4_k \).

To simplify the explanations, only the coding of the luminance component is described below. However, the same approach can be used for coding the chrominance components.

Starting from the image \( X_{DCT} \), a probabilistic distribution \( P \) of each DCT coefficient is determined using a parametric probabilistic model. This step is referenced 190 in the Figure.

Since, in the present example, the image \( X_{DCT} \) is a residual image, i.e. information is about a noise residual, it is efficiently modelled by Generalized Gaussian Distributions (GGD) having a zero mean: \( DCT(X) \approx GGD(\alpha, \beta) \),

where \( \alpha, \beta \) are two parameters to be determined and the GGD follows the following two-parameter distribution:

\[
GGD(\alpha, \beta, x) = \frac{\beta}{2\alpha \Gamma(1/\beta)} \exp\left(\frac{-|x/\alpha|^\beta}{\beta}\right),
\]
and where $\Gamma$ is the well-known Gamma function: $\Gamma(\zeta) = \int_0^\infty t^{\zeta - 1}e^{-t}dt$.

The DCT coefficients cannot be all modelled by the same parameters and, practically, the two parameters $\alpha, \beta$ may depend on:

- video content. This means that the parameters must be computed each image or every $n$ images for instance;

- the index $i$ of the DCT coefficient within a DCT block $B_k$. Indeed, each DCT coefficient has its own behaviour. A DCT channel is thus defined as the DCT coefficients collocated (i.e. having the same index) within a plurality of DCT blocks (possibly all the blocks of the image). A DCT channel can therefore be identified by the corresponding index $i$; and/or

- the encoding mode used for the collocated block of the base layer, referred in the present document as to the "base coding mode". Typically, Intra blocks of the base layer do not behave the same way as Inter blocks. Blocks with a coded residual in the base layer do not behave the same way as blocks without such a residual (i.e. Skipped blocks). And blocks coded with a non-nil texture data according to the coded-block-pattern syntax element as defined in H.264/AVC do not behave the same way as those blocks without non-nil texture data.

It is to be noted that, due to the down-sampling of the base layer, the collocation of blocks should take into account that down-sampling. For example, the four blocks of the $n$-th macroblock in the residual enhancement layer with UHD resolution are collocated with the $n$-th block of the base layer having a HD resolution. That is why, generally, all the blocks of a macroblock in an enhancement image have the same base coding mode.

For illustrative purposes, if the residual enhancement image $X_{DCT}$ is divided into 8x8 pixel blocks, the modelling 190 has to determine the parameters of 64 DCT channels for each base coding mode.

In addition, since the luminance component $Y$ and the chrominance components $U$ and $V$ have dramatically different source contents, they must be encoded in different DCT channels. For example, if it is decided to encode the luminance component $Y$ on one channel and the chrominance components $UV$ on another channel, 128 channels are needed for each base coding mode.

At least 64 pairs of parameters for each base coding mode may appear as a substantial amount of data to transmit to the decoder (see parameters 21). However, experience proves that this is quite negligible compared to the volume of data needed
to encode the residuals of Ultra High Definition (4k2k or more) videos. As a consequence, one may understand that such a technique is preferably implemented on large videos, rather than on very small videos because the parametric data would be too costly.

For the sake of simplicity of explanation, a set of DCT blocks corresponding to the same base coding mode is now considered. The invention may then be applied to each set corresponding to each base coding mode. Furthermore, as suggested above, the invention may be directly applied to the entire image, regardless of the base coding modes.

To obtain the two parameters $\alpha$, $\beta$, defining the probabilistic distribution $P$, for a DCT channel $i$, the Generalized Gaussian Distribution model is fitted onto the DCT block coefficients of the DCT channel, i.e. the DCT coefficients collocated within the DCT blocks with the same base coding mode. Since, this fitting is based on the values of the DCT coefficients before quantization (of the DCT blocks having the same base coding mode in the example), the probabilistic distribution is a statistical distribution of the DCT coefficients within a considered channel $i$.

For example, the fitting may be simply and robustly obtained using the moment of order $k$ of the absolute value of a GGD:

$$M_k^{\alpha_i, \beta_i} := \mathbb{E} \left( \left| GGD(\alpha_i, \beta_i) \right|^k \right)_{(k \in R_+)}$$

$$= \int_{-\infty}^{\infty} |x|^k GGD(\alpha_i, \beta_i, x) \, dx = \frac{\alpha_i^k \Gamma(1 + k / \beta_i)}{\Gamma(1 / \beta_i)}.$$

Determining the moments of order 1 and of order 2 from the DCT coefficients of channel $i$ makes it possible to directly obtain the value of parameter $\beta$:

$$\frac{M_1}{(\mathbb{W}_i)^2} = \frac{\Gamma(1 / \beta_i) \Gamma(3 / \beta_i)}{\Gamma(2 / \beta_i)^2}$$

The value of the parameter $\beta$, can thus be estimated by computing the above ratio of the two first and second moments, and then the inverse of the above function of $\beta$.

Practically, this inverse function may be tabulated in a memory of the encoder instead of computing Gamma functions in real time, which is costly.

The second parameter $\alpha$, may be determined from the first parameter $\beta$, and the second moment, using the equation: $M_2 = \sigma^2 = \alpha_i^2 \frac{\Gamma(3 / \beta_i)}{\Gamma(1 / \beta_i^2)}$. 
The two parameters $\alpha_i, \beta_i$ being determined for the DCT coefficients $i$, the probabilistic distribution $P_i$ of each DCT coefficient $i$ in a considered block is defined by

$$P_i(x) = \text{GGD}(a_{1f}, \beta_i, x_i) = \frac{\beta_i}{2\alpha_i \Gamma(1/\beta_i)} \exp(-|x_i|/\alpha_i^{\beta_i}).$$

Still referring to Figure 3, a quantization of 192 of the DCT coefficients is then performed, to obtain quantized DCT coefficients (i.e. symbols or values).

As shown in the Figure, the quantization of those coefficients may involve optimal quantizers chosen (step 191) for each DCT channel $i$ based on the corresponding probabilistic distribution $P_j(x)$ of the DCT coefficients.

In a variant, the quantizers may be predefined prior to the encoding. Since the quantization is not the core of the present invention, it is here assumed that a quantizer is selected for each DCT channel and each base coding mode as defined above, meaning that various quantizers are generally used for quantizing various DCT coefficients.

Figure 7 illustrates an exemplary Voronoi cell based quantizer.

A quantizer is made of $M$ Voronoi cells distributed over the values of the DCT coefficients. Each cell corresponds to an interval $[f_m, f_{m+1}]$, called quantum $Q_m$.

Each cell has a centroid $c_m$, as shown in the Figure.

The intervals are used for quantization: a DCT coefficient comprised in the interval $[f_m, f_{m+1}]$ is quantized by a symbol $a_m$, associated with that interval.

The centroids are used for de-quantization: a symbol $a_m$ associated with an interval is de-quantized into the centroid value $c_m$ of that interval.

Figure 8 illustrates what is named an alphabet of quantization for a given quantizer.

This Figure shows an example of symbols $a_m$ corresponding to the Voronoi cells. In practice, those symbols are consecutive integer values centred on 0, i.e. values from $[-\text{MAX}; \text{MAX}]$ as shown in the Figure.

The alphabet $A$ comprises all the symbols or values $a_m$, defined by the Voronoi cells of the quantizer used. In other words, the alphabet $A$ comprises all the possible symbols or values for a DCT coefficient due to the quantization performed on that coefficient.

Considering the quantizer used for each DCT coefficient in the DCT image $X_{\text{DCT}}$, an alphabet $A_i$ is obtained for each DCT coefficient $i$. This is schematically
illustrated by Figure 9, in which \( A_i \) denotes the alphabet available for a \( i \)-th DCT coefficient within the DCT blocks \( B_k \).

One may note that the alphabets of two DCT coefficients belonging to the same DCT channel (i.e. collocated DCT coefficients from different blocks \( B_k \)) are identical, since the same quantizer is used. An illustrative example is given in Figure 9 where the four blocks \( B \) of a macroblock \( MB_k \) use the same alphabets \( A_1, A_2, A_3, \) etc. for their quantized DCT symbols.

It is to be recalled here that due to different coding modes of the base layer macroblocks, different kinds of alphabets, indexed by the corresponding base coding mode, are used to code the residual macroblocks (i.e. according to the corresponding DCT channel).

However, it is to be recalled here that the explanations below concentrate on macroblocks associated with the same coding mode, i.e. using the same alphabets. This is because the residual enhancement image may be segmented according to the base coding modes, and then processed by segment each corresponding to the same coding mode.

Still referring to Figure 3, based on the probabilistic distribution \( P_i(x) \) associated with a DCT channel \( i \) (for a base coding mode), the probabilities \( p_{im} \) of occurrence of each symbol or value \( a_m \) of the corresponding alphabet \( A_i \) are calculated.

This may be done by computing the integral \( \int_{b_m} P_i(x)\,dx \) on the quantum \( Q_m \).

The probabilities \( \{p_{im}\} \) for an alphabet \( A_i \) thus define the probabilistic distribution of the possible symbols or values defined therein.

Given the considered DCT channel \( i \) for a base coding mode, the probabilistic distribution is the same for the alphabets associated with DCT coefficients collocated within a plurality of the blocks of the image.

Such probabilities may be computed off-line and stored in memory of the encoder, in order to decrease the complexity of real time encoding. This is for example possible when the parameters \( \alpha, \beta \) for modelling the distribution of the DCT coefficients are chosen from a limited number of possible parameters, and when the possible quantizers are known in advance.

The probabilities \( \{p_{im}\} \) of each alphabet \( A_j \) and the quantized symbols or values \( a_m \) obtained from each DCT coefficient in the DCT image \( X_{DCT} \) are then provided for entropy coding 193 as shown in Figure 3. In this respect, the quantized DCT coefficients are generally processed according to a zigzag scan.
Figure 3a gives more details about the present invention. Before generating binary entropy codes for the possible values or symbols of each alphabet, some processing of those alphabets may be performed in order to optimize their entropy when entropy encoding the DCT coefficients. The same references are used in Figures 3 and 3a.

Two additional processing operations are illustrated on the Figure, namely the restriction 194 of alphabets and the grouping 195 of alphabets. Those two processing operations can be implemented separately and one without the other.

As will become apparent below, the restriction 194 of alphabets leads to generating an additional stream with the encoded bit-stream EBS, namely the magnitude probability bit-stream 22. The grouping 195 of alphabets does not need such an additional stream.

The second processing operation, i.e. the grouping of alphabets before entropy encoding the DCT coefficients based on the resulting grouped alphabets, is explained below.

The restriction of alphabets 194 is an extension of the Skip flag as introduced in H.264 for example. It results in restricted alphabets that well fit to the encoding of the DCT blocks, given the maximum magnitude of their quantized DCT coefficients.

Similarly to the Skip flag of H.264, a flag is associated with each block of quantized DCT coefficients.

That flag for a DCT block specifies a magnitude, which may be determined based on the quantized coefficient values in the associated block. The magnitude is the extended information compared to conventional Skip flags.

And thanks to that extended information, the encoding method allows restriction of alphabets A, corresponding to the quantized DCT coefficients of each block, based on the magnitude specified in the flag associated with that block.

By restricting the alphabets, the number of values therein is reduced, also reducing the entropy of the corresponding binary codes. This is because the bits that would be required for enabling the encoding of unused values (those that are deleted thanks to the restriction of alphabets) are avoided.

More details about a first implementation are now given.

In this first implementation, the magnitude specified in a flag associated with a block represents a maximum magnitude of the quantized coefficient values of that block. This specified maximum magnitude may be the maximum magnitude taken
by one of the quantized coefficient values, but may also be an upper bound of those values.

Let's define the probability \( PB_n \) that, in a given block \( B \), the maximum magnitude of the quantized DCT coefficients is \( n : P_{B_n} := P(\max_m \abs{a_m} \geq n) \) where \( a_m \) are the quantized DCT coefficients (in fact their values or symbols). These probabilities are computed after the quantization step based on the values of the quantized DCT coefficients, once the encoder knows the quantized value of each DCT coefficient.

Let's also define, for a macroblock \( MB \), the probability \( P_{MB} \) that the maximum magnitude of the quantized DCT coefficients of the (four) blocks of macroblock \( MB \) is \( n \). These probabilities \( P_{MB,n} \) are computed by the encoder after the quantization process, based on the actual state of the macroblocks (either from the entire image or from the part of the image corresponding to the same base coding mode).

Due to the correlation between neighbouring blocks, it appears that \( P_{MB,0} = P_{B,0}^4 \), and more generally that \( P_{MB,n} = P_{B,n}^4 \).

Figure 21 illustrates an embodiment of the main steps of the restriction.

The method begins at step 2100 with the alphabets \( A_i \) of the quantized DCT coefficients considered (e.g. from the blocks associated with the same base coding mode) and their associated probabilistic distributions \( \{P_{im}\} \).

Step 2105 consists in computing the probabilities \( P_{B,n} \) as defined above.

This enables definition of a new alphabet \( F \) of the possible flags \( f_n \) for blocks, each flag \( f_n \) being associated with the probability \( P_{B,n} \).

Thus, the alphabet \( F \) comprises the flags \( f_n \), where \( n \) defines the maximum magnitude (i.e. the maximum absolute value) of the quantized DCT coefficients \( a_m \) in a block. In other words, the alphabet \( F \) comprises the flags specifying, as maximum magnitude, all the possible values defined in the alphabets \( A \).

At step 2110, a flag \( f_n \) from the alphabet \( F \) is associated with each of the DCT blocks \( B_k \), based on the maximum magnitude \( n \) taken by a quantized DCT coefficient therein. Those flags specify the maximum magnitudes taken by one of the quantized coefficient values of their respective associated block. Those flags will be entropy encoded as further described below.

As a variant, this step may follow step 2115 and then a flag from the restricted alphabet \( F' \) as defined below may be selected and directly associated with
each DCT block rather than a flag from the larger alphabet F. In that case, the flags may specify only an upper bound of the magnitude taken by the quantized coefficient values of their respective associated block.

The next step 2115 consists in restricting this set or alphabet F of flags into a restricted alphabet F', based on probabilities of occurrence $P_{A_n}$ of the flags $f_n$ of F.

This is because, since the block flag $f_n$ may take many different values for a block (i.e. the alphabet F may be large), the encoding cost of the flag may turn out bigger than the overhead reduction obtained by using the Huffman coding based on the restricted alphabets $A_i$ as defined below.

The alphabet F is thus restricted to the most significant flag values, which will be used to restrict the alphabet $A_i$, as further described below.

An example of selecting the most significant flag values is now given, even though other strategies may be implemented in the context of the invention.

In particular, the number of flags selected to constitute the restricted alphabet $F'$ is at most three. This appears to be a good tradeoff between the costs of encoding the flags and of encoding the DCT coefficients based on the restricted alphabets $A_i$, as defined below.

Furthermore, in order to be able to losslessly encode all quantized DCT blocks, one flag of the flags in the restricted alphabet $F'$ is the flag specifying the largest magnitude in the alphabet F.

Let's denote the three values by a, b and $\omega$, with $a < b < \infty$ and $\omega$ standing for the largest magnitude of the alphabet F: thus $F' = \{a, b, \omega\}$.

The values a and b may be selected to define intervals of magnitudes (i.e. [0,a], [a,b]) corresponding to a probability of occurrence greater than a predefined amount, for example 10%. The probability of occurrence is the probability that a block has a maximum magnitude comprised in [0,a] or [a,b].

For example, a may be chosen as the smallest value with at least 10% probability of occurrence, and thus computed based on the following formula:

$$a = \min \{ t | 10% \leq P(n \leq t) \text{, where the block flag is } f_n \}$$

where $P(n \leq t)$ means the probability of occurrence of a block having a maximum magnitude equal to or less than t. $P(n \leq t) = \sum_{n \leq t} P_{B,n}$. 
Similarly, the other value \( b \) may be chosen as the smallest value with the next at least 10% probability of occurrence from the previous value \( a \). \( b \) verifies the following formula:

\[
b = \min \left\{ i \mid 10\% \leq P(a < n \leq t) \text{ where the block flag is } f_n \right\}
\]

where \( P(a < n \leq t) = \sum_{0 \leq t \leq 1} P_{b,n} \).

One may observe that such conditions do not necessary lead to three different values \( a < b < \infty \). They may result in a restricted alphabet \( F' \) having only two values or even only one value \( f_0 \) if all blocks are zero-skipped or \( f_{\infty} \) if only very few blocks do not reach the largest magnitude.

In a variant where only two values \( a \) and \( \infty \) are considered, a predefined amount of 40% may be used to determine \( a \).

The conventional Skip flag corresponds to \( f_0 \). In that case, one flag in the restricted alphabet \( F \) specifies a (maximum) magnitude equal to zero.

Following step 2115, step 2120 consists in restricting the alphabets \( A \), corresponding to each quantized DCT coefficient in a current block based on the (maximum) magnitude specified in the flag associated with the current block to process.

In particular, the restricting operation is as follows:

- all blocks whose flag value is \( f_{-n} \), with \( n \leq a \) are restricted based on the flag value \( f_a \);
- all blocks whose flag value is \( f_n \) with \( a < n \leq b \) are restricted based on the flag value \( f_b \);
- all blocks whose flag value is \( f_{\infty} \), with \( b < n \) are restricted based on the flag value \( f_{\infty} \).

Restricting the alphabet \( A \) means calculating the intersection between that alphabet and the interval \([-n,n]\) corresponding to the associated flag \( l \). \([A \cap [-n,n]]\). In other words, the values of \( A \) that correspond to a magnitude larger than \( n \) are deleted.

In that case, the probabilities \( \{P_{l,n}\} \) associated with each possible values are normalized given that restriction, in order to enable further entropy coding.

One may note that the alphabets with all values less than \( n \) are not restricted but kept as such.

However, as an alternative to restricting the alphabet \( A \), all the alphabets corresponding to the quantized DCT coefficients of a given block are replaced by a
single alphabet made of values equal to the maximum magnitude specified in the flag
associated with that block. Thus, the same alphabet is used when encoding all the
quantized DCT coefficients of the block, possibly requiring the building of only one
Huffman tree.

As shown in Figure 3a, the probabilities \( P_{B,n} \) and \( PMB^* \) (n=a,b,∞), as well
as the values a, b and ∞, for the restricted alphabet \( F^* \) are included in the bit-stream 22,
where the probabilities \( P_{B,n} \) for the restricted alphabet are the sums of the previous
probabilities \( P_{B,n} \) over respectively the intervals \([0,a], [a,b] \) and \([b,∞]\).

Sometimes, \( PMB^* \) with \( n ≠ 0 \) do not really need to be transmitted as is
apparent from examples below.

In a second implementation of the flags, the distance of DCT coefficient
magnitudes to the maximum possible magnitude allowed by the quantizers (when they
may be determined) is used. In this implementation, the magnitude specified in a flag
associated with a block represents the value \( M_B d_B \), where \( M_B \) is the maximum
magnitude of the possible values of all the alphabets and \( d_B \) is a block distance to the
maximum magnitude defined by \( d_B = \min_{i m, a_m ≠ 0} M_i - |a_m| \) with \( a_m \) representing the
quantized coefficient values of the block, \( A \) representing one of the alphabets, \( M_i \) being
the maximum magnitude of the possible values of the alphabet \( A_i \), and \( d_B = M_B \) when
all the quantized coefficient values \( a_m \) of the block are zero.

Let's define \( M_{r,i} \) as the maximum value of the alphabet \( A_i \) of the quantizer
of the \( i \)-th DCT coefficient; this means that values taken by this quantizer are in the
interval \([-M_{r,i}, M_{r,i}]\).

Let \( M_B \) be the maximum possible magnitude of the blocks: \( M_B = \max_i M_i \).

The block distance to the maximum magnitude is defined as:

\[
d_B = \min_{i m, a_m ≠ 0} M_i - |a_m|
\]

\( a_m \in A_i \)

and \( d_B = M_B \) if all DCT coefficients are zero.

The flag \( f_n \) can therefore be defined for each block, where \( n = M_B - d_B \)
(which is referred to below as a "plane level").
One can see that the common skipped block full of zeroes corresponds to a nil plane level \( n = 0 \).

A plane level \( n = 1 \) means that quantizers have been planed \( M_B - 1 \) times; one iteration of planing means that the two outermost quanta have been dropped, until only the central 0 quantum remains.

A plane level \( n = i \) means that quantizers have been planed \( M_B - i \) times. In particular, \( n = M_B \) means that quantizers have not been planed.

Figure 22 illustrates such plane levels for the quantizers of six DCT coefficients.

In this Figure, restricting the alphabets of a block comprises deleting the \( d_B \) most negative and \( d_B \) most positive possible values from each of the alphabets. This corresponds to dropping the \( d_B \) outermost quanta from each alphabet, until the sole central quantum remains.

Similarly to the above, the probabilities \( P_B \) and \( P_MB \) may then be defined (respectively as the probability that the plane level of a given block is equal to \( n \), and as the probability that the plane level of a whole macroblock is equal to \( n \)).

A similar alphabet \( F \) as above is also obtained, which can be restricted as also explained above.

In a variant to \( n = M_B - d_B \), \( n \) may be equal to \( d_B \) as defined above. This has the same result except that \( n = 0 \) does not correspond to the skip mode as defined in H.264.

Based on the restricted alphabets obtained at step 2125 (when the restricting operation 194 is implemented) or on the initial alphabets \( A_j \) (otherwise), and their associated probabilities, the grouping 195 may be applied.

The probabilistic distributions (i.e. the probabilities \( p_{ikmp} \)) of the alphabets are used to group at least two alphabets into a new alphabet, the new alphabet replacing the at least two alphabets and defining, as symbols, the combinations of possible symbols for the at least two transformed block coefficients associated with the grouped alphabets.

The grouping of alphabets is to be understood as a product of those alphabets, as set out for example in the book "Elements of Information Theory" (T.M. Cover, J.A. Thomas, Second Edition., ed. Wiley, 2006).
For instance, the product or grouping of two alphabets \( A = \{a_i\} \) and \( B = \{b_j\} \) is made of all the pairs of symbols \( c_{ij} = (a_i, b_j) \), with which a corresponding probability is associated \( p(c_{ij}) = p(a_i)p(b_j) \).

The idea here is to consider together several DCT symbols resulting from the quantization as a new single symbol for which a corresponding binary code will be generated using an entropy coding scheme. It appears that the binary codes generated from the new alphabet (resulting from the grouping) give better mean length relative to each alphabet that is grouped.

This improvement is due to the fact that the variable-length-code (VLC) binary codes generated from an alphabet alone have an integer number of bits, moving away its mean length from the theoretical entropy of the alphabet (up to one bit). On the other hand, the product of several alphabets makes it possible to dilute this one bit cost into several alphabets, decreasing the mean length associated with the part of binary codes for each alphabet.

For illustrative purposes, without grouping of alphabets, the VLC code of each alphabet is not worse than one bit (the above "one bit cost") from the entropy and satisfies the following inequalities: \( H(A) \leq L(A) < H(A) + 1 \), where the entropy \( H(A) \) is defined by \( H(A) = -\sum p_i \log_2 p_i \), and the mean length of the VLC code is defined by

\[
L(A) = \sum_i p_i L_i \quad (L_i \text{ being integer in terms of number of bits of the generated binary code associated with the symbol } a_i).
\]

When grouping several alphabets, let say \( N \) times the alphabet \( A : B = A^N \), the above inequalities for \( B \) are:

if we consider that there is no dependence between occurrences of symbols of the alphabet \( A \) (which is the worst case scenario), we have

\[
Z(B) < H(B) + 1 = NH(A) + 1.
\]

Since \( H(B) \leq L(B) < H(B) + 1 \), it may be inferred that \( L'(A) = L(B) / N < H(A) + 1 / N \), proving that the mean length associated with the part of binary codes for each alphabet (when the binary codes are generated for the product of the alphabets) may be substantially improved.

In theory, by taking \( N \) large, the entropy encoding may be made as close as wanted to the theoretical entropy, but this would not be practical since large Huffman trees would then have to be handled by the encoder.
Different strategies for grouping the alphabets may be implemented as described below, providing VLC binary codes that are closer, to a greater or lesser extent, to the theoretical entropy of the alphabets.

**Figure 10** illustrates an overall method of processing the alphabets associated with the quantized DCT symbols, according to an embodiment of the invention.

The method begins at step 1000 with the alphabets A of the quantized DCT symbols considered (e.g. from the blocks associated with the same base coding mode) and their associated probabilistic distributions \( \{P_{i,m}\} \).

At step 1005, which is optional, alphabets having associated entropy or a size (in terms of number of symbols) less than a predefined value are discarded from consideration for grouping alphabets.

For example, the very small alphabets A with \( H(A) < 0.05 \) bit are discarded. These alphabets can be dropped because their impact on the PSNR is negligible.

Optionally, when an alphabet is discarded, the encoding method may avoid generating corresponding VLC codes (e.g. Huffman trees) as described below, and may avoid encoding the DCT symbols coming from this alphabet.

Next, at step 1010, it is decided to group alphabets according to an intra-block grouping. It is recalled here that the grouping of alphabets is a product of those alphabets.

Such intra-block grouping groups at least two alphabets associated with at least two DCT coefficients of the same block of pixels.

Such situation is illustrated in **Figure 11**, in which a first grouping groups the alphabets \( A_1 \) and \( A_2 \) for the DCT coefficients \( i=1 \) and \( i=2 \) of the same DCT block \( B_k \). A similar grouping is made with the alphabets \( A_3, A_5 \) and \( A_0 \).

In the example of this Figure, the at least two DCT coefficients for which the grouping is performed are adjacent coefficients within the same block.

Given that several DCT blocks are using the same alphabets, when it is decided to perform an intra-block grouping, the same grouping of alphabets may be performed for a plurality of blocks within the first image, in particular the blocks of the same macroblock (because they are associated with the same base coding mode).

**Figure 11** illustrates this situation for two adjacent blocks \( B_k \) and \( B_{k+1} \).

More details of the intra-block grouping, in particular regarding the criteria to select alphabets to group, are given below with reference to **Figure 13**.
Further to the intra-block grouping, step 1015 is an inter-block grouping of the alphabets remaining after step 1010, i.e. they may be initial alphabets resulting from the quantization and/or new alphabets resulting from the intra-block grouping.

Such inter-block grouping groups at least two alphabets associated with DCT coefficients collocated in at least two blocks of pixels. In particular, the at least two blocks of pixels may be from the same macroblock dividing the DCT image $X_{DCT}$.

This is illustrated in Figure 12, in which the intra-block grouping of $A_3$, $A_5$ and $\alpha \beta$ in the four blocks of the macroblock $MB_k$ are also grouped to form a new alphabet $(A_3 \times A_5 \times \alpha \beta)^4$.

More details of the inter-block grouping, in particular regarding the criteria to select alphabets to group, are given below with reference to Figure 14.

It can be seen that a new alphabet resulting from intra or inter-block grouping replaces the alphabets that are grouped, and that a probabilistic distribution of that new alphabet is also computed as defined above (i.e. based on the probabilistic distributions of the at least two alphabets involved in the grouping). This is because such probabilistic distribution of the new alphabet may be useful when considering whether or not that new alphabet should be grouped again, but is also required to generate VLC codes.

At the end of step 1015, a set of alphabets and associated probabilistic distributions is obtained (step 1020).

Intra-block grouping and then inter-block grouping are performed iteratively as long as a resulting new alphabet comprises less symbols than a predefined number. That means that the iterative process is continued up to a given upper limit of the cardinal of the grouped alphabets. This is in order to ensure that the associated VLC codes, e.g. conventional Huffman trees, are small enough to be handled in real time by the codec.

Several methods for the intra-block grouping will now be discussed.

With reference to Figure 13, a first method for intra-block grouping of alphabets is based on the entropy of those alphabets. This is because the alphabets with very small entropy are those with the largest overhead, due to the upper bound of one bit that may be asymptotically reached. On the contrary, the alphabets with high entropy have generally a lower overhead.

Given this, the method comprises determining an entropy value for each of the alphabets, and the grouping of at least two alphabets comprises grouping the available alphabet having the smallest entropy with the available alphabet having the
highest entropy. As known, the entropy value derives from the probabilistic distribution of the alphabet.

When it is desired to ensure that the resulting new alphabet comprises less symbols than a predefined number, the smallest (and respectively the largest) entropy alphabet may not be the alphabet having the smallest (and respectively the largest) entropy as such, but those alphabets that satisfy this constraint. For example, the alphabet with the smallest entropy is chosen for a grouping with the compatible (i.e. alphabet product size is smaller than the given upper limit) alphabet having the highest entropy.

The grouping based on the probabilistic distributions (indirectly the entropy) of the alphabets is iteratively repeated until there are no more alphabets satisfying the constraint. It is to be noted that a group of alphabets resulting from the intra-block grouping is considered as an alphabet for the next intra-block grouping iteration.

The initial state 1300 comprises the alphabets $A_1, \ldots, A_n$ of a DCT block that is to be processed, as well as their associated probabilistic distribution $\{p_{i,m}\}$, for $m \geq 0$ ($mi$ being an index varying from 0 to the number of symbols within the corresponding alphabet $A_j$). To each alphabet $A_i$ is appended the list of the corresponding DCT coefficients, initially only the DCT coefficient with index i.

At step 1305, the entropy $H(A_i)$ of each alphabet is computed, and the alphabet $A_{\text{min}}$ having the minimum entropy is selected.

At step 1310, the alphabet $A_{\text{max}}$ which has the maximum entropy amongst the other alphabets, while satisfying the constraint on the cardinal of the product $A_{\text{min}} \times A_{\text{max}}$ (cardinal less that an upper limit denoted MAX_ALPHABET_SIZE in the Figure), is then selected.

In case such a pair $(A_{\text{min}}, A_{\text{max}})$ is found (test 1315), the step 1320 is executed during which:

- the product $A_{\text{min}} \times A_{\text{max}}$ is considered as a new alphabet for the next iteration, by determining its corresponding symbols $c_i$. $A_{\text{min}}$ and $A_{\text{max}}$ taken alone are discarded from the list of the alphabets for the next iteration;

- the probabilistic distribution $\{p_{\text{new,m}}\}$ of the new alphabet is calculated based on the probabilistic distribution of the grouped alphabets: $p_{\text{new,m}} = p_{\text{min,m}} \times p_{\text{max,m}}$;

- a list of the DCT coefficients associated with the new alphabet is appended to that alphabet.
In case no pair is found at test 1315, the intra-block grouping ends at step 1325 with a plurality of alphabets (some group initial alphabets $A_j$) and their associated probabilistic distributions.

Such intra-block grouping efficiently improves the entropy of VLC codes generated from the resulting alphabets. This is because in the resulting alphabet, the largest probability (which induces a large overhead) is bounded by the highest probability of the alphabet with high entropy. Therefore, the remaining highest probability induces reduced overhead.

A second method for intra-block grouping of alphabets is now described. Compared to the first method, this method is more efficient but requires more calculation. Instead of only calculating the entropy of the alphabets, the second method selects alphabets such that the reduction of coding overhead is maximum, meaning that the steps 1305 and 1310 are different (but the other steps are the same).

To that end, the second method comprises grouping two alphabets which maximizes a reduction in distance between the entropy of the alphabets and the entropy of the binary codes constructed from the alphabets. Generally, the binary codes are generated from the alphabet and the associated probabilistic distribution, based on a Huffman code.

Unfortunately, the distance of a Huffman code to the entropy cannot be anticipated without a costly and impractical operation of constructing the Huffman tree itself. In order to simplify the computational complexity, the entropy of the binary codes constructed from the alphabets is modelled by the entropy of a corresponding Shannon code. This is because the distance $d(A_j)$ as defined below provides an upper bound of the Huffman code distance to the entropy of the alphabet $A_j$.

The entropy of Shannon code is defined by $\sum_r \log \left( \frac{1}{\bar{p}_r} \right)$ where $\pi$ is the integer ceiling function.

In the second method, this entropy is used to estimate the overhead or distance to the entropy of the alphabet. This distance of the Shannon code to the entropy of the alphabet $A$, is defined as follows:

$$d(A_j) = \left[ \sum_{r=1} p_i \pi \left( \log \left( \frac{1}{p_i} \right) \right) \right] - \left( \sum_{s=1} p_j \log \left( \frac{1}{p_j} \right) \right).$$

The reduction in overhead provided by the grouping of two alphabets $A_i$ and $A_j$ is the following:
Reduction^ = d(A_i) + d(A_j) - d(A_i, A_j).

Therefore, the second method for the intra-block coding consists in selecting (corresponding to the two steps 1305 and 1310) the two alphabets A_i and A_j which provide the maximum reduction together while satisfying the constraint on cardinality:

\[
(A_i, A_j) = \underset{\text{card}(A_i, A_j) \leq \text{MAX}_\text{ALPHABET}_\text{SIZE}}{\text{ArgMax}} \{ d(A_i) + d(A_j) - d(A_i, A_j) \}
\]

If such a pair is found, step 1320 is performed, ensuring that the group of alphabets resulting from the intra-block grouping is considered as an alphabet for the next intra-block grouping iteration, with an associated probabilistic distribution.

As for the first method, the grouping of the alphabets is iteratively repeated until there are no more alphabets satisfying the constraint on the size of A_i, A_j.

A third method for the intra-block grouping derives from the second method, wherein each distance relating to an alphabet is weighted by the entropy of that alphabet. This gives priority to grouping alphabets having small entropy rather than alphabets with larger entropy, when those alphabets have binary codes with similar distances to the entropy.

Compared to the second method, the alphabets to group are selected based on the following formula:

\[
(A_i, A_j) = \underset{\text{card}(A_i, A_j) \leq \text{MAX}_\text{ALPHABET}_\text{SIZE}}{\text{ArgMax}} \left\{ \frac{d(A_i)}{H(A_i)} + \frac{d(A_j)}{H(A_j)} - \frac{d(A_i, A_j)}{H(A_i, A_j)} \right\}
\]

Whatever the method selected for the intra-group grouping, step 1010 ends with a set of M alphabets (some grouping several initial alphabets A_i) and associated probabilistic distributions.

**Figure 14** illustrates an example of inter-block grouping that may be applied to these alphabets obtained at the end of step 1010. Let's denote those alphabets by A'_k, when beginning the inter-block grouping 1015.

In this example, the inter-block grouping is performed between blocks B_k of the same macroblock MB_k. Such a situation has been previously introduced with reference to **Figure 12**.

The inter-block grouping is performed iteratively by making the product alphabet of an alphabet A'_k, multiplied by itself, due to the same base coding mode within the macroblock (which permits the same intra-block grouping to be performed in all the blocks of that macroblock).

If it is grouped zero times, the result is one alphabet per block B_k.
If it is grouped once, the result is one alphabet per 2 blocks (i.e. half the macroblock).

If it is grouped twice, the result is one alphabet per 4 blocks (i.e. per a macroblock). This is the case of Figure 12 in which the alphabet A'=(A_3 \times A_5 \times A^\beta) is grouped twice to obtain the new alphabet \((A_3 \times A_5 \times A^\beta)^4\) for the macroblock MB_k. In this example, while \((A_3 \times A_5 \times A^\beta)^4\) is only one alphabet, its symbols represent 12 DCT coefficients at the same time, namely DCT coefficients with indexes 3, 5 and 6 of the four blocks \(B_k^1, B_k^2, B_k^3\) and \(B_k^4\).

With reference to Figure 14, the initial state 1400 comprise the M alphabets \(A'_\alpha\), as well as their associated probabilistic distributions \(\{Pi_{m}\}\). nbCoeffs defines the number of DCT coefficients that are appended to a considered alphabet.

Step 1405 initialises an index a enabling the iterative processing of the inter-block grouping through steps 1425 and 1430. Each iteration a processes the alphabet \(A'_\alpha\), trying to group it with itself, once or twice (output "no" of test 1420 driving the second grouping of this alphabet).

In the example, the only condition for allowing a grouping is the given upper limit (MAX_ALPHABET_SIZE) in the cardinality on the inter-block grouped alphabet. However, other conditions may be implemented. They are checked at step 1410.

When a grouping of the alphabet \(A'_\alpha\) with itself is possible, step 1415 is performed. In particular, this step updates the appended list of DCT coefficients concerned by the new alphabet obtained. This list doubles each time the alphabet is grouped with itself.

Step 1415 also updates the probabilistic distribution of that new alphabet.

This iterative processing is over when all the alphabets \(A'_\alpha\) have been processed, meaning that they have been grouped twice or they cannot satisfy the inter-block grouping constraint regarding cardinality.

Step 1015 of inter-block grouping ends with a set of M' alphabets (some grouping several initial alphabets \(A_i\) for several blocks) and associated probabilistic distributions.

Such a set is obtained for a specific macroblock with which a specific base coding mode is associated, as introduced above.

Following the grouping, binary entropy codes for the symbols defined in the M' alphabets are generated, in such a way that the quantized DCT symbols are then encoded using those generated binary codes.
The same order of the DCT symbols is applied at the encoding and at the decoding, since the decoder performs the same operations as the encoder (grouping based on the probabilities and probabilistic distribution parameters that it will receive). For example, this order may be the zigzag scanning order, taking into account the grouping of alphabets (i.e. grouping of DCT coefficients): the smallest index of a DCT coefficient amongst the grouped indexes may be used to represent the encoding/decoding position of the grouped coefficients, along the zigzag order.

The binary codes may be VLC codes obtained from the probabilistic distribution, based on a Huffman code. Huffman trees are thus generated to obtain the corresponding VLC binary codes for each of the M' alphabets.

The entropy coding of the quantized DCT coefficients of a macroblock is illustrated in Figure 15. The process shown in this Figure is based on the fact that it is possible to alternate between many Huffman trees during the encoding without altering the decoding capability: the generated bit-stream 20 is still a prefix code and there is no need to add extra syntax to specify which DCT coefficients of the blocks have been grouped within a single alphabet (i.e. which DCT coefficients are encoded by the same binary code), since the decoder will perform the same grouping operations and obtain the same grouped alphabets.

The algorithm of Figure 15 has to be processed for each macroblock of the DCT image $X_{dct}$ to encode the entire residual enhancement image.

The process begins (step 1500) with the obtaining of the restricted flag alphabet $F'$ as constructed above, based on the current base coding mode.

At step 1502, the flags $f_s$, $f_b$, or $f_r$, depending on which interval $[0,a]$, $[a,b]$ or $]b,\infty[$ the value $n$ belongs to. This means that one of $f_s$, $f_b$ and $f_r$, is thus associated with each block of the current macroblock. This substitution between $f_s$ and one of $f_r$, $f_b$ and $f_k$ can however be done previously, when restricting the alphabets.

In particular, the flag associated with each block is entropy encoded based on the probabilities of occurrence of the flags of the alphabet $F$, used to construct VLC (e.g. Huffman) codes.

However, to further improve the encoding of those flags, a grouping as defined above can be implemented: the flags associated with the blocks of the same macroblock are concatenated into one flag associated with the macroblock. And the flag associated with the same macroblock are entropy encoded based on probabilities of occurrence of the possible values for such a macroblock flag $f_{MB}$, those possible
values resulting from the product of the alphabet $F'$ of flags from which the concatenated flags are selected: $f_{MB} \in F_{MB}^e := F \times F \times F \times F$.

Furthermore, the probabilities of occurrence of the possible values for a macroblock flag are calculated based on the probability $P_{MB,n}$ defined above. This is because since $p_{MB,n} = p_{MB}$, the other probabilities for a macroblock flag have to be normalized. For example, if only $p_{MB,0}$ is considered for that normalization:

\[ p(f_0, f_0, f_0, f_0) = p_{MB,0} \]
\[ p(f_i, f_i, f_i, f_i) = y \cdot p_{B,i} \cdot p_{B,j} \cdot p_{B,k} \]

with $(i, j, k, l) \in \{a, b, \ldots\}^4$ and is not $(0,0,0,0)$

and $y = \left(1 - p_{MB,0}\right)/ \sum_{(i, j, k, l) \neq 0} p_{B,i} \cdot p_{B,j} \cdot p_{B,k} \cdot p_{B,l}$ normalize the probabilities.

In a variant the probabilities $P_{MB,n}$ may also be used, in which case $y$ is adapted: $y = \left(1 - \sum_n p_{MB,n}\right)/ \sum_{(i, j, k, l) \neq 0} p_{B,i} \cdot p_{B,j} \cdot p_{B,k} \cdot p_{B,l}$ and the probabilities $p_{MB,n}$ are provided in the bit-stream to the decoder.

For each macroblock, the macroblock flag $f_{MB}$ is thus encoded using a Huffman coding based on the above probabilities.

Figures 23 illustrates the order of coding: a flag $f_{MB}$ followed by the Huffman coding of its associated macroblock data, then another flag followed by the Huffman coding of the associated next macroblock and so on.

Such a structure preserves the spatial random access capability in the stream.

It is to be noted that if a block has the flag $f_0$ associated with it, the encoding of that block is skipped because the decoder will know from that flag that all values are zeroes in that block.

Further to step 1502, step 1504 consists in obtaining the set $S$ of $M'$ symbol groups (i.e. of the $M'$ alphabets issued from the intra and inter-block grouping) corresponding to that coding mode.

For the following explanations:
- $g$ represents the current alphabet whose symbols are being encoded for the current macroblock;
- $g.nbDCTCoeffs$ represents the total number of DCT coefficients coded by each symbol of the current alphabet $g$ (i.e. the number of DCT coefficients appended to that alphabet);
- \( g.n_b\) Block represents the value \( nb\) Block obtained during the inter-block grouping for the alphabet \( g \) (see Figure 14);
- \( g.f\) actor, represents the cardinal of individual alphabets \( A_j \) that were grouped together to form the current alphabet \( g \); and

5  \[ \text{quantized\_coeff}[g, \text{dct}\_i + \text{block\_index} \times 64] \] is the value of the \( i^{\text{th}} \) quantized DCT coefficient stored in the list appended to the current alphabet \( g \), for the current symbol (since a symbol may represent several initial DCT coefficients). The current symbol being encoded is represented by the \( \text{block\_index} \) variable.

The \( M^- \) alphabets are processed iteratively through steps 1505, 1545 (which detects the last alphabet) and 1550 (which selects another alphabet).

For the current alphabet \( g \), the algorithm encodes one or more symbols of that alphabet, depending on the number of 8x8 DCT blocks (variable \( nb\) Block obtained from Figure 14 for that alphabet) associated with the current alphabet.

For instance, if the current alphabet \( g \) results from two inter-block groupings (i.e. \( A_j \) has been grouped with itself twice), then one symbol of that alphabet encodes the appended DCT coefficients in the four blocks of the current macroblock \((nb\) Block=4).

If it results from only one inter-block grouping, then it encodes the appended DCT coefficients of two consecutive 8x8 blocks \((nb\) Block=2).

If it has not been inter-block grouped, then it encodes the appended DCT coefficients of each block of the macroblock \((nb\) Block).

Therefore, the number of symbols coded for the current alphabet depends on the number of blocks it covers. Thus, a loop is performed to encode the symbols of that current alphabet \( g \), through steps 1510 (which initializes \( \text{block\_index} \)), 1530 (which increments the \( \text{block\_index} \) according to \( nb\) Block) and 1535 (which checks whether all the blocks of the macroblock have been processed).

For each symbol to code (i.e. occurring in the macroblock), the symbol value \( \text{symbol\_Val} \) is computed at step 1515. This consists in combining quantized DCT coefficient values together so as to obtain the right symbol value.

30  Once the symbol value \( \text{symbol\_Val} \) is obtained, the Huffman code associated with that symbol value is obtained at step 1520 from the constructed Huffman trees. Then that Huffman code is written in the output video bitstream 20 at step 1525.

Thanks to the use of the flags \( f(n) \) (or \( f(M) \)), the DCT coefficients are encoded using the restricted alphabets. In particular:
- all blocks whose flag value is $f_n$, with $n < a$ are encoded with flag value $f_a$, i.e. with alphabets having magnitudes less than $a$ (for the first implementation of the flag as described above);
- all blocks whose flag value is $f_n$ with $a < n \leq b$ will be encoded with flag value $f_b$, i.e. with alphabets having magnitudes between $a$ and $b$;
- all blocks whose flag value is $f_n$ with $b < n$ will be encoded with flag value $f_n'$, i.e. without restricting the alphabets.

As explained above, this reduces the average number of bits (entropy) to encode the DCT coefficients. One may note that it does not greatly increase the computational complexity since at most two extra Huffman trees will be constructed (associated with $a$ and $b$).

For the sake of completeness, with reference to Figure 16, the corresponding decoding of the bitstream 20 will now be described, as schematically shown in Figure 4a, where 333 references the restricting of alphabets thanks to the flags $f_n$ for decoding, and 334 references the grouping of alphabets for decoding.

The decoding is to be performed successively for each encoded macroblock.

This process is similar to that of Figure 15 regarding the loops on the alphabets and on the block index.

One difference between the coding and the decoding algorithm consists in step 1602 which is the decoding of the macroblock flag $F_{M_0}$ and step 1604 which includes obtaining the alphabets restricted given the flags $f_n$.

During this decoding of the flags, the process comprises
- obtaining alphabets of values for the encoded quantized DCT coefficients, each alphabet defining the possible values of a quantizer associated with an encoded quantized coefficient;
- obtaining, from the bit-stream, a flag associated with each block of encoded quantized coefficients, said flag specifying a magnitude;
- restricting alphabets corresponding to the encoded quantized coefficients of each block, based on the magnitude specified in the flag associated with that block.

In particular, from the bit-stream 22, probabilities are obtained associated with each of the alphabet $F'$ of possible flags; and the encoded flags are entropy decoded from the bit-stream based on the obtained probabilities to obtain said flag.
associated with each block. Those probabilities make it possible to reconstruct the
Huffman trees. They are the probabilities associated with the flags \( f_a, f_b \) and \( f_\infty \).

As mentioned above, the macroblock flag concatenates several block flags. Therefore, the decoding process comprises splitting the decoded macroblock flag into
flags associated with the blocks of the macroblock.

Thanks to the decoded flags and the known quantizers, it is possible to
determine the restricted alphabets \((A, \Pi \ [-n, n])\), based on which the decoding of the
encoded quantized coefficients is performed.

In particular, the decoder performs the same restriction as the one
performed by the encoder:
- with the first implementation of the flags, restricting the alphabets of a block
comprises deleting, from the alphabets, the possible values that have a magnitude
(i.e. absolute value) larger than the magnitude specified in the flag associated with
the block;
- with the second implementation (the flag conveys information about block distance
to the maximum magnitude, referred to as \( d_B \)), restricting the alphabets of a block
comprises deleting the \( d_B \) most negative and \( d_B \) most positive possible values from
each of the alphabets. In case the magnitude \( n \) specified in the flag equals to \( M_B - d_B \), it is to be noted that \( M_B \) can be retrieved from the probabilistic distributions of
the DCT coefficients (i.e. from parameters \( \alpha, \beta \)) and the associated quantizers, to
deduce \( d_B \).

Another difference between the coding and the decoding algorithm consists
in the two decoding steps illustrated by the dashed lines in the Figure.

These two steps consist in the decoding of a Huffman code, through the
use of the Huffman tree associated with current grouped alphabet \( g \). The decoded
Huffman code provides a symbol of the alphabet \( g \).

This symbol then provides the values of the quantized DCT coefficients that
are associated with current alphabet \( g \), in a way symmetrical to step 1515. In particular,
the decoding may comprise:

- obtaining, from the bit-stream EBS, blocks of encoded symbols \( a_m \) corresponding to
quantized block DCT coefficients;
- obtaining alphabets \( A \) of symbols, each alphabet defining the possible symbols of
an associated quantized block coefficient;
- for each alphabet, obtaining a probabilistic distribution of the possible symbols
defined therein;
using the probabilistic distributions to group at least two alphabets into a new alphabet, the new alphabet defining, as symbols, the combinations of possible symbols of the at least two quantized block coefficients associated with the grouped alphabets;

- obtaining binary entropy codes for the possible symbols of each remaining alphabet;
- decoding the encoded symbols using the obtained binary codes.

More particularly, parameters $(\alpha_i, \beta_i)$ of each alphabet $A_i$ can be obtained from the bit-stream $21$ and then applied to the above defined probabilistic distribution model $\text{GGD}(\alpha, \beta)$ to obtain the probabilistic distribution of each alphabet (requiring integrating that model on the considered quantums $Q_m$ as explained above). Given the knowledge of the quantizers by the decoder, the latter is then able to perform the same grouping operations as the encoder, resulting in the same grouped alphabets.

The improvement of the entropy coding thanks to the grouping of alphabets is illustrated in Figure 17, which is similar to Figure 5 in which a curve of the rate-distortion performance obtained with the grouping according to the invention is shown (dotted curve).

The Figure shows that a bitrate close to the theoretical entropy rate is achieved due to the intra and inter-block grouping of the invention.

The compression gain resulting from the use of the above flags $f_n$ is about 5%, when both the restriction of alphabets and the entropy encoding of those flags are implemented.

The entropy coding scheme according to the present embodiment also has spatial random access properties, when the video encoding process is without inter frame (temporal) and intra block (spatial) predictions.

This is because the entropy coding according to the present embodiment has no dependence between macroblocks. In particular, if an entry point of the generated bit-stream $20$ and the index of the associated macroblock are given, it is possible to perform the entropy decoding from that point, without decoding other parts of the encoded video.

It is said that the bit-stream has the random spatial access property because it is possible to decode only a part of the image (a region of interest) once the associated entry points are given.
**Figure 18** illustrates how the residual enhancement image may be subdivided into spatial zones made of macroblocks, with entry points in order to allow efficient random access compliant coding.

The position of the entry points may be encoded in the header of the bit-stream 20 in order to facilitate easy extraction from the server side and allow the reconstruction of a valid stream on the decoder side.

**Figure 19** shows the meta-organization of a bit-stream header. For example, the slice header shown in the Figure re-uses the slice header of the H.264/AVC video compression standard. However, to provide the entry points, there is added, at the beginning of each slice header, a new field ("coded slice length") which indicates the length in bytes of the coded slice. The entry points can therefore be easily computed from the "coded slice length" fields.

Another advantage of this independence between macroblocks of the residual enhancement image is the possibility to perform parallel entropy decoding on the decoder side. Each decoding thread starts decoding from one of the entry points as defined above.

This invention provides a simple, efficient and block based entropy coder of the quantized DCT coefficients. In addition, the possible absence of intra prediction between blocks and the judicious choice of the entropy coder provides random spatial access into the video bitstream, i.e. the bitstream can be decoded from any entry point in the bit-stream.

**Example Encoding/Decoding Apparatus**

With reference now to **Figure 20**, a particular hardware configuration of a device for encoding or decoding images able to implement methods according to the invention is now described by way of example.

A device implementing the invention is for example a microcomputer 50, a workstation, a personal digital assistant, or a mobile telephone connected to various peripherals. According to yet another embodiment of the invention, the device is in the form of a photographic apparatus provided with a communication interface for allowing connection to a network.

The peripherals connected to the device comprise for example a digital camera 64, or a scanner or any other image acquisition or storage means, connected to an input/output card (not shown) and supplying image data to the device.

The device 50 comprises a communication bus 51 to which there are connected:
- a central processing unit CPU 52 taking for example the form of a microprocessor;
- a read only memory 53 in which may be contained the programs whose execution enables the methods according to the invention. It may be a flash memory or EEPROM;
- a random access memory 54, which, after powering up of the device 50, contains the executable code of the programs of the invention necessary for the implementation of the invention. As this memory 54 is of random access type (RAM), it provides fast access compared to the read only memory 53. This RAM memory 54 stores in particular the various images and the various blocks of pixels as the processing is carried out (transform, quantization, storage of the reference images) on the video sequences;
- a screen 55 for displaying data, in particular video and/or serving as a graphical interface with the user, who may thus interact with the programs according to the invention, using a keyboard 56 or any other means such as a pointing device, for example a mouse 57 or an optical stylus;
- a hard disk 58 or a storage memory, such as a memory of compact flash type, able to contain the programs of the invention as well as data used or produced on implementation of the invention;
- an optional diskette drive 59, or another reader for a removable data carrier, adapted to receive a diskette 63 and to read/write thereon data processed or to process in accordance with the invention; and
- a communication interface 60 connected to the telecommunications network 61, the interface 60 being adapted to transmit and receive data.

In the case of audio data, the device 50 is preferably equipped with an input/output card (not shown) which is connected to a microphone 62.

The communication bus 51 permits communication and interoperability between the different elements included in the device 50 or connected to it. The representation of the bus 51 is non-limiting and, in particular, the central processing unit 52 unit may communicate instructions to any element of the device 50 directly or by means of another element of the device 50.

The diskettes 63 can be replaced by any information carrier such as a compact disc (CD-ROM) rewritable or not, a ZIP disk or a memory card. Generally, an information storage means, which can be read by a micro-computer or microprocessor, integrated or not into the device for processing a video sequence, and which may
possibly be removable, is adapted to store one or more programs whose execution permits the implementation of the method according to the invention.

The executable code enabling the coding device to implement the invention may equally well be stored in read only memory 53, on the hard disk 58 or on a removable digital medium such as a diskette 63 as described earlier. According to a variant, the executable code of the programs is received by the intermediary of the telecommunications network 61, via the interface 60, to be stored in one of the storage means of the device 50 (such as the hard disk 58) before being executed.

The central processing unit 52 controls and directs the execution of the instructions or portions of software code of the program or programs of the invention, the instructions or portions of software code being stored in one of the aforementioned storage means. On powering up of the device 50, the program or programs which are stored in a non-volatile memory, for example the hard disk 58 or the read only memory 53, are transferred into the random-access memory 54, which then contains the executable code of the program or programs of the invention, as well as registers for storing the variables and parameters necessary for implementation of the invention.

It will also be noted that the device implementing the invention or incorporating it may be implemented in the form of a programmed apparatus. For example, such a device may then contain the code of the computer program(s) in a fixed form in an application specific integrated circuit (ASIC).

The device described here and, particularly, the central processing unit 52, may implement all or part of the processing operations described in relation with Figures 1 to 19 and 21 to 34, to implement methods according to the present invention and constitute devices according to the present invention.

Channel Modelling and Selection of Quantizers

The following description focuses on channel modelling and selection of quantizers.

The quality of a video or still image may be measured by the so-called Peak-Signal-to-Noise-Ratio or PSNR, which is dependent upon a measure of the L2-norm of the error of encoding in the pixel domain, i.e. the sum over the pixels of the squared difference between the original pixel value and the decoded pixel value. It may be recalled in this respect that the PSNR may be expressed in dB as:

$$10 \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right)$$

where MAX is the maximal pixel value (in the spatial domain) and...
MSE is the mean squared error (i.e. the above sum divided by the number of pixels concerned).

However, as noted above, most of video codecs compress the data in the DCT-transformed domain in which the energy of the signal is much better compacted.

The direct link between the PSNR and the error on DCT coefficients is now explained.

For a residual block, we note $\psi_\eta$ its inverse DCT (or IDCT) pixel base in the pixel domain as shown on Figure 24. If one uses the so-called IDCT III for the inverse transform, this base is orthonormal: $\|\psi_\eta\| = 1$.

On the other hand, in the DCT domain, the unity coefficient values form a base $\phi_\eta$ which is orthogonal. One writes the DCT transform of the pixel block $X$ as follows:

$$X_{ocr} = \sum_n d^n \phi_n,$$

where $d^n$ is the value of the n-th DCT coefficient. A simple base change leads to the expression of the pixel block as a function of the DCT coefficient values:

$$X = \text{IDCT}(X_{ocr}) = \text{IDCT}\sum_n d^n \phi_n = \sum_n d^n \text{IDCT}(\phi_n) = \sum_n d^n \psi_n .$$

If the value of the de-quantized coefficient $d''$ after decoding is denoted $d''_n$, one sees that (by linearity) the pixel error block is given by:

$$\varepsilon_x = \sum_n (d'' - d''_n) \psi_n$$

The mean $L_2$-norm error on all blocks, is thus:

$$E(\|\varepsilon_x\|_2^2) = E\left(\sum_n |d'' - d''_n|^2\right) = \sum_n E\left(|d'' - d''_n|^2\right) = \sum_n D_n^2$$

where $D_n^2$ is the mean quadratic error of quantization on the n-th DCT coefficient, or squared distortion for this type of coefficient. The distortion is thus a measure of the distance between the original coefficient (here the coefficient before quantization) and the decoded coefficient (here the dequantized coefficient).

It is thus proposed below to control the video quality by controlling the sum of the quadratic errors on the DCT coefficients. In particular, this control is preferable compared to the individual control of each of the DCT coefficient, which is a priori a sub-optimal control.
In the embodiment described here, it is proposed to determine (i.e. to select in step 191 of Figure 3) a set of quantizers (to be used each for a corresponding DCT channel), the use of which results in a mean quadratic error having a target value $D_i^2$ while minimizing the rate obtained.

In view of the above correspondence between PSNR and the mean quadratic error $D_n^2$ on DCT coefficients, these constraints can be written as follows:

$$\min R = \sum_n R_n(D_n) \quad \text{s.t.} \quad \sum_n R_n(D_n) = D_i^2 \quad (A)$$

where $R$ is the total rate made of the sum of individual rates $R_n$ for each DCT coefficient. In case the quantization is made independently for each DCT coefficient, the rate $R_n$ depends only on the distortion $D_n$ of the associated n-th DCT coefficient.

It may be noted that the above minimization problem (A) may only be fulfilled by optimal quantizers which are solution of the problem

$$\min R_n(D_n) \quad \text{s.t.} \quad \mathbb{E}\left[|d^* - d_n|^2\right] = \sum_n D_n^2 \quad (B).$$

This statement is simply proven by the fact that, assuming a first quantizer would not be optimal following (B) but would fulfill (A), then a second quantizer with less rate but the same distortion can be constructed (or obtained). So, if one uses this second quantizer, the total rate? has been diminished without changing the total distortion $\sum_n D_n^2$; this is in contradiction with the first quantifier being a minimal solution of the problem (A).

As a consequence, the rate-distortion minimization problem (A) can be split into two consecutive sub-problems without losing the optimality of the solution:

- first, determining optimal quantizers and their associated rate-distortion curves $R_n(D_n)$ following the problem (B), which will be done in the present case for GGD channels as explained below;

- second, by using optimal quantizers, the problem (A) is changed into the problem $(A_{opt})$:

$$\min R = \sum_n R_n(D_n) \quad \text{s.t.} \quad \sum_n D_n^2 = D_i^2 \quad \text{and} \quad R_n(D_n) \text{ is optimal} \quad (A_{opt}).$$

Based on this analysis, it is proposed as further explained below:

- to compute off-line optimal quantizers adapted to possible probabilistic distributions of each DCT channel (thus resulting in the pool of quantizers of Figure 3);
- to select one of these pre-computed optimal quantizers for each DCT channel (i.e. each type of DCT coefficient) such that using the set of selected quantizers results in a global distortion corresponding to the target distortion $D^2_i$ with a minimal rate (i.e. a set of quantizers which solves the problem $A_{opt}$).

It is now described a possible embodiment for the first step of computing optimal quantizers for possible probabilistic distributions, here Generalised Gaussian Distributions.

It is proposed to change the previous complex formulation of problem (B) into the so-called Lagrange formulation of the problem: for a given parameter $\lambda > 0$, we determine the quantization in order to minimize a cost function such as $D^2 + XR$. We thus get an optimal rate-distortion couple $(D_\lambda, R_\lambda)$. In case of a rate control (i.e. rate minimisation) for a given target distortion $\Delta$, the optimal parameter $\lambda > 0$ is determined by $\lambda_{\delta} = \arg\min_{\lambda, D_\lambda \leq \Delta} (i.e. the value of $\lambda$ for which the rate is minimum while fulfilling the constraint on distortion) and the associated minimum rate is $R_{\delta} = R_{\lambda_{\delta}}$.

As a consequence, by solving the problem in its Lagrange formulation, for instance following the method proposed below, it is possible to plot a rate distortion curve associating a resulting minimum rate to each distortion value $(\Delta, \leftrightarrow \Delta_{\lambda_{\delta}})$ which may be computed off-line as well as the associated quantization, i.e. quantizer, making it possible to obtain this rate-distortion pair.

It is precisely proposed here to formulate problem (B) into a continuum of problems (BJambda) having the following Lagrange formulation

$$\text{minimize } D^2_{\lambda} + \lambda R_{\lambda}(D_{\lambda}) \quad \text{s.t. } E|X - dm|^2 = D^2_{\lambda} \quad \text{(BJambda).}$$

The well-known Chou-Lookabaugh-Gray algorithm is a good practical way to perform the required minimisation. It may be used with any distortion distance $d$; we describe here a simplified version of the algorithm for the $L^2$-distance. This is an iterative process from any given starting guessed quantization.

As noted above, this algorithm is performed here for each of a plurality of possible probabilistic distributions (in order to obtain the pre-computed optimal quantizers for the possible distributions to be encountered in practice), and for a plurality of possible numbers $M$ of quanta. It is described below when applied for a given probabilistic distribution $P$ and a given number $M$ of quanta.
In this respect, as the parameter alpha $\alpha$ (or equivalently the standard deviation $\sigma$ of the Generalized Gaussian Definition) can be moved out of the distortion parameter $D^2_\sigma$ because it is a homothetic parameter, only optimal quantizers with unity standard deviation $\sigma = 1$ need to be determined in the pool of quantizers.

Taking advantage of this remark, in the proposed embodiment, the GGD representing a given DCT channel will be normalized before quantization (i.e. homothetically transformed into a unity standard deviation GGD), and will be de-normalized after de-quantization. Of course, this is possible because the parameters (in particular here the parameter $\alpha$ or equivalently the standard deviation $\sigma$) of the concerned GGD model are sent to the decoder in the video bit-stream.

Before describing the algorithm itself, the following should be noted. The position of the centroids $c_m$ is such that they minimize the distortion $\delta^2_m$ inside a quantum, in particular one must verify that $d_{c_m}\delta^2_m = 0$ (as the derivative is zero at a minimum).

As the distortion $\delta_m$ of the quantization, on the quantum $Q_m$, is the mean error $E(d(x;c_m))$ for a given distortion function or distance $d$, the distortion on one quantum when using the $L^2$-distance is given by $\delta_m^2 = \int_{Q_m} |x - c_m|^2 P(x)dx$ and the nullification of the derivative thus gives: $c_m = \int_{Q_m} xP(x)dx/P_m$, where $P_m$ is the probability of $x$ to be in the quantum $Q_m$ and is simply the following integral

$$P_m = \int_{Q_m} P(x)dx.$$

Turning now to minimisation of the cost function $C = D^2 + \lambda R$, and considering that the rate reaches the entropy of the quantized data:

$$R = 2 \sum_{m=1}^{M} P_m \log_2 P_m,$$

the nullification of the derivatives of the cost function for an optimal solution can be written as:

$$0 = \partial_{m+1} C = d_{m+1}(A^2_m - \lambda P_m \ln P_m + A^2_{m+1} - \lambda P_{m+1} \ln P_{m+1}).$$

Let us set $P = P(t_{m+1})$ the value of the probability distribution at the point $t_{m+1}$. From simple variational considerations, see Figure 25, we get

$$\partial_{m+1} P_m = \bar{P}$$

and

$$\partial_{m+1} P_{m+1} = -\bar{P}.$$
Then, a bit of calculation leads to
\[
\partial_{t_{m+1}} \Delta_m^2 = \partial_{t_{m+1}} \int_a^b |x - c_m|^2 P(x) \, dx \\
= \overline{P}'|t_{m+1} - c_m|^2 + \int_a^b \partial_{t_{m+1}} |x - c_m|^2 P(x) \, dx \\
= \overline{P}'|t_{m+1} - c_m|^2 - 2\partial_{t_{m+1}} c_m \int_a^b (x - c_m) P(x) \, dx \\
= \overline{P}'|t_{m+1} - c_m|^2
\]
as well as
\[
\partial_{t_{m+1}} \Delta_{-m+1} = -\overline{P}'|t_{m+1} - c_{m+1}|^2.
\]
As the derivative of the cost is now explicitly calculated, its cancellation gives:
\[
0 = \overline{P}'|t_{m+1} - c_m|^2 - \lambda \overline{P}' \ln P_m - \lambda P_m \frac{\overline{P}'}{P_m} - \overline{P}'|t_{m+1} - c_{m+1}|^2 + \lambda \overline{P}' \ln P_{m+1} + \lambda P_m \frac{\overline{P}}{P_m},
\]
which leads to a useful relation between the quantum boundaries \( t_{m+1} \) and the centroids \( c_m \):
\[
t_{m+1} = \frac{c_m + c_{m+1}}{2} - \frac{\lambda \ln P_{m+1} - \ln P_m}{2(c_{m+1} - c_m)}.
\]
Thanks to these formulae, the Chou-Lookabaugh-Gray algorithm can be implemented by the following iterative process:

1. Start with arbitrary quanta \( Q_m \) defined by a plurality of limits \( t_m \)
2. Compute the probabilities \( P_m \) by the formula \( P_m = \int_{Q_m} P(x) \, dx \)
3. Compute the centroids \( c_m \) by the formula \( c_m = \int_{Q_m} xP(x) \, dx / P_m \)
4. Compute the limits \( t_m \) of new quanta by the formula
\[
t_{m+1} = \frac{c_m + c_{m+1}}{2} - \frac{\lambda \ln P_{m+1} - \ln P_m}{2(c_{m+1} - c_m)}
\]
5. Compute the cost \( C = D^2 + \lambda R \) by the formula \( C = \sum_{m=1}^M \Delta_m^2 + \lambda P_m \ln P_m \)
6. Loop to 2. until convergence of the cost \( C \)

When the cost \( C \) has converged, the current values of limits \( t_m \) and centroids \( c_m \) define a quantization, i.e. a quantizer, with \( M \) quanta, which solves the problem (BJambda), i.e. minimises the cost function for a given value \( \lambda \), and has an associated rate value \( R_\lambda \) and an distortion value \( D_\lambda \).
Such a process is implemented for many values of the Lagrange parameter λ (for instance 100 values comprised between 0 and 50). It may be noted that for λ equal to 0, there is no rate constraint, which corresponds to the so-called Lloyd quantizer.

In order to obtain optimal quantizers for a given parameter β of the corresponding GGD, the problems (BJambda) are to be solved for various odd (by symmetry) values of the number M of quanta and for the many values of the parameter λ. A rate-distortion diagram for the optimal quantizers with varying M is thus obtained, as shown on Figure 26.

It turns out that, for a given distortion, there is an optimal number M of needed quanta for the quantization associated to an optimal parameter λ. In brief, one may say that optimal quantizers of the general problem (B) are those associated to a point of the upper envelope of the rate-distortion curves making this diagram, each point being associated with a number of quanta (i.e. the number of quanta of the quantizer leading to this point of the rate-distortion curve). This upper envelope is illustrated on Figure 27. At this stage, we have now lost the dependency on λ of the optimal quantizers: for a given rate (or a given distortion) corresponds only one optimal quantizer whose number of quanta M is fixed.

Based on observations that the GGD modelling provides a value of β almost always between 0.5 and 2 in practice, and that only a few discrete values are enough for the precision of encoding, it is proposed here to tabulate β every 0.1 in the interval between 0.2 and 2.5. Considering these values of β (i.e. here for each of the 24 values of β taken in consideration between 0.2 and 2.5), rate-distortion curves, depending on β, are obtained as shown on Figure 28. It is of course possible to obtain according to the same process rate-distortion curves for a larger number of possible values of β.

Each curve may in practice be stored in the encoder in a table containing, for a plurality of points on the curve, the rate and distortion (coordinates) of the point concerned, as well as features defining the associated quantizer (here the number of quanta and the values of limits \( t_m \) and centroids \( c_m \) for the various quanta). For instance, a few hundreds of quantizers may be stored for each β up to a maximum rate, e.g. of 5 bits per DCT coefficient, thus forming the pool of quantizers mentioned in Figure 3. It may be noted that a maximum rate of 5 bits per coefficient in the
enhancement layer makes it possible to obtain good quality in the decoded image. Generally speaking, it is proposed to use a maximum rate per DCT coefficient equal or less than 10 bits, for which value near lossless coding is provided.

Before turning to the selection of quantizers, for the various DCT channels and among these optimal quantizers stored in association with their corresponding rate and distortion when applied to the concerned distribution (GGD with a specific parameter $\beta$), it is proposed here to possibly encode only part of the DCT channels.

Based on the observation that the rate decreases monotonously as a function of the distortion induced by the quantizer, precisely in each case in the manner shown by the curves just mentioned, it is possible to write the relationship between rate and distortion as follows:

$$R_n = f_n(-\ln(D_n^i \sigma_n)),$$

where $\sigma_n$ is the normalization factor of the DCT coefficient, i.e. the GGD model associated to the DCT coefficient has $\sigma_x$ for standard deviation, and where $f_n' \geq 0$ in view of the monotonicity just mentioned.

In particular, without encoding (equivalently zero rate) leads to a quadratic distortion of value $\sigma_n^2$ and we deduce that $0 = f_n'(0)$.

Finally, one observes that the curves are convex for parameters $\beta$ lower than two: $f \leq 2 \implies f'' \geq 0$

It is proposed here to consider the merit of encoding a DCT coefficient. More encoding basically results in more rate $R_n$ (in other words, the corresponding cost) and less distortion $D_n^i$ (in other words the resulting gain or advantage).

Thus, when dedicating a further bit to the encoding of the video (rate increase), it should be determined on which DCT coefficient this extra rate is the most efficient. In view of the analysis above, an estimation of the merit $M$ of encoding may be obtained by computing the ratio of the benefit on distortion to the cost of encoding:

$$M := \frac{\Delta D^2}{\Delta R^2}.$$

Considering the distortion decreases by an amount $\varepsilon$, then a first order development of distortion and rates gives

$$(D - \varepsilon)^2 = D^2 - 2sD + o(s)$$

and
$$R(D - \varepsilon) = f_n(-\ln((D-e)/a))$$
$$= f_n(-\ln(D/a) - \ln(-e/D))$$
$$= f_n(-\ln(D/\sigma) + \varepsilon I D + O(\varepsilon))$$
$$= f_n(-\ln(D/\sigma)) + ef'(-\ln(D/a))/D$$

As a consequence, the ratio of the first order variations provides an explicit formula for the merit of encoding:
$$M_n(D_n) = \frac{2D^2}{f_n'(-\ln(D/\sigma))}.$$
The exact number of DCT channels to be encoded is determined during the quantizer selection process as explained below.

The optimality of this solution (encoding only the first coefficients ordered by initial merit) can be proven as follows.

If we assume that a first given DCT coefficient \( n_j \) is encoded and that there is another second DCT coefficient \( n_j \) which is not encoded and which has a higher initial merit, then an infinitesimal amount of coding rate can be taken from the first coefficient \( n_j \) to encode the second coefficient \( \alpha_j \). Because one has

\[
M_{n_j}^0 \geq M_{n_j}^0 \geq M_{n_j} (D_n).
\]

it is clear that one gets a lower distortion for the same rate. So, the encoding of the DCT coefficients is not optimal and one understands that, if a coefficient is encoded, then all coefficients with higher initial merits must be encoded.

As a corollary, if there are \( N \) DCT coefficients (or channels) per block, the number of possible configurations that should be envisaged when deciding which coefficients to encode drops from \( 2^N \) (decision for each coefficient whether it should be encoded) to \( N + 1 \) (after ordering by decreasing initial merit, the number of coefficient may vary from 0 to \( N \)).

The encoding priority just mentioned does not specify whether a DCT coefficient is more or less encoded than another DCT coefficient; it indicates however that, if a DCT coefficient is encoded at a non-zero rate, then all coefficients with higher priority must be encoded at a non-zero rate.

The encoding priority provides an optimal encoding order that may be compared to the non-optimal conventional zigzag scan coding order used in MPEG, JPEG, H.264 and HEVC standard video coding.

Based on the pre-computed optimal quantizers determined above and the possible sets of DCT channels to be encoded as just explained, it is possible to solve the optimisation problem \( (A_{opt}) \), \( i.e. \) to select one of these pre-computed optimal quantizers for each DCT channel to be encoded such that using the set of selected quantizers results in a global distortion corresponding to the target distortion \( D_t^2 \) with a minimal rate as follows. (This selection step corresponds to the choice referenced 191 in Figure 3.)
The domain of optimization is as shown on Figure 29. The quality constraint $\sum_n D^2_n = D^2_i$ can be rewritten as $h = 0$ with $h(D_1, D_2, \ldots) := \sum_n D^2_n - D^2_i$.

The distortion of each DCT coefficient is upper bounded by the distortion without coding: $D_n \leq \sigma_n$, and the domain of definition of the problem is thus a multidimensional box $\Omega = \{(z_r, D_2, \ldots); D_n \leq \sigma_n\} = \{(D_r, D_2, \ldots); g_n \leq 0\}$, defined by the functions $g_n(D_n) := D_n - \sigma_n$.

Thus, the problem can be restated as follows:

**minimize** $R(D, D_2, \ldots)$  s.t. $h = 0, g_n \leq 0$ (A_opf).

Such an optimization problem under inequality constrains is for instance solved using so-called Karush-Kuhn-Tucker (KKT) necessary conditions of optimality.

In this goal, the relevant KKT function $\Lambda$ is defined as follows:

$$A(D_1, D_2, \ldots, \lambda, \mu_1, \mu_2, \ldots) := R - \lambda h - \sum_n \mu_n g_n.$$  

The KKT necessary conditions of minimization are

- stationarity: $dA = 0$,
- equality: $h = 0$,
- inequality: $g_n \leq 0$,
- dual feasibility: $\mu_n \geq 0$,
- saturation: $\mu_n g_n = 0$.

It may be noted that the parameter $\lambda$ in the KKT function above is unrelated to the parameter $\lambda$ used above in the Lagrange formulation of the optimisation problem meant to determine optimal quantizers.

If $g_n = 0$, the $n$-th condition is said to be saturated. In the present case, it indicates that the $n$-th DCT coefficient is not encoded.

By using the specific formulation $R_n = f_n(-\ln(D_n, 1, \sigma_n))$ of the rate depending on the distortion discussed above, the stationarity condition gives:

$$0 = d_\lambda A = d_\lambda R_n = - d_\lambda \ln(D_n, 1) = - d_\lambda f_n' / D_n = 2 \lambda D_n - \mu_n,$$

i.e. $2 \lambda D_n^2 = - \mu_n D_n - f_n'$.  

By summing on $n$ and taking benefit of the equality condition, this leads to

$$2 \lambda D_i^2 = - \sum_n \mu_n D_n - \sum_n f_n'. (*)$$
In order to take into account the possible encoding of part of the coefficients only as proposed above, the various possible indices \( n \) are distributed into two subsets:

- the set \( I^0 = \{ \eta; \mu_\eta = 0 \} \) of non-saturated DCT coefficients (i.e. of encoded DCT coefficients) for which we have \( \mu_\eta D_\eta = 0 \) and \( D_\eta^2 = -f_\eta' / 2\lambda \), and

- the set \( I^+ = \{ n; \mu_\eta > 0 \} \) of saturated DCT coefficients (i.e. of DCT coefficients not encoded) for which we have \( \mu_\eta D_\eta = -f_\eta' \lambda \sigma_\eta^2 \).

From (*) we deduce

\[
2\lambda D_i^2 = -\sum \mu_\eta D_\eta - \sum f_\eta' = \sum f_\eta' + 2\lambda \sum \sigma_\eta^2 - \sum f_\eta',
\]

and by gathering the \( \lambda \)'s

\[
2\lambda \left( D^2_i - \sum \sigma_\eta^2 \right) = \sum f_\eta'.
\]

As a consequence, for a non-saturated coefficient \( (n \in I^0) \), i.e. a coefficient to be encoded, we obtain:

\[
D_\eta^2 = \left( D^2_i - \sum \sigma_\eta^2 \right) f_\eta'(-\ln(D_\eta / \sigma_\eta)) / \sum f_m'(-\ln(D_m / \sigma_m)).
\]

It may be noted that this is an implicit system of equations because the derivatives \( f_\eta' \) depend on the distortions \( D_\eta \). It is proposed to solve numerically this implicit system by a fixed point algorithm.

For a given set \( I^0 \) of non-saturated coefficients, the above system can be rewritten \( \vec{D} = \vec{F}(\vec{D}) \), using a continuous vector function \( \vec{F} \) and with \( \vec{D} = (D_1, D_2, ...) \). A fixed point method may be used to solve such a system by defining a series \( \vec{D}(t + 1) = \vec{F}(\vec{D}(t)) \) and \( \vec{D}(0) \) arbitrary among the possible solutions (in the sub-space of the box \( \Omega \) with dimensions corresponding to the set \( I^0 \)). If this series converges to a limit \( \vec{D}(\infty) \), by continuity of the function \( \vec{F} \), this limit is solution of the problem.

It may be noted in addition that, by theorem, the series converges if the function \( \vec{F} \) is a contracting function, i.e. if its differentiae is smaller than one. As this is not always the case, it is possible to force the convergence using a penalization
method: the fixed point problem $\bar{D} = \tilde{F}(\bar{D})$ can be rewritten as another fixed point problem:

$$
\bar{D} = \bar{G}(\bar{D}) = \theta \tilde{F}(\bar{D}) + (1-\theta)\bar{D}.
$$

By taking the parameter $\Theta$ close to zero, one can force the differentiate of $G$ to be as close to one as wanted, ensuring the contraction of $G$. However use of very small $\theta$'s lead to a very slow convergence and a balance may thus be found in practice.

In view of the above, the practical algorithm for solving the implicit system defined above is as follows:

1. For each non-saturation set $I^o$ of the $N+1$ possible non-saturation sets (provided by the priority of encoding as explained above), the iterative fixed point method is performed:

   1. if $D^2 + \sum_{I^o} \sigma_n^2 < 0$, encoding is impossible with the concerned non-saturation set $I^o$;
   2. start with arbitrary distortions $D_n(0)$ (for $n \in I^o$), for example with $A_n(0) = \sigma_n$;
   3. determine the distortions $D_n(t+1)$ (for $n \in I^o$) thanks to the formula:

$$
D_n^{t+2} = \left(D^2 + \sum_{I^o} \sigma_n^2\right) f_n'(-\ln(D_n(t)/\sigma_n)) + \sum_{m \in I^o} f_m'(-\ln(D_m(t)/\sigma_m))
$$

and the penalization: $D_n(t+1) = \theta D_n^t + (1-\theta)G D_n(t)$

for a fixed parameter $\Theta$. Compute the associated rate $R(t+1) = \sum_{n \in I^o} R_n(t+1)$, where $R_n(t+1)$ is the rate associated with the distortion $D_n(t+1)$;

4. loop on 3. until convergence of the rates $R(t+1)$, and store the final rate under $R_{f^o}$;

II. determine the minimum rate $R_{\text{min}}$ among all rates $R_{f^o}$ (taken into account the the $N+1$ possible non-saturation sets $I^o$). The optimal DCT distortions $D_n$ (for values of $n$ belonging to the set $I^o$ for which the minimum rate is obtained) are those associated with this minimum rate and were determined during the execution of the
previous algorithm (1 above). For each DCT channel to be encoded, the selected quantizer is the quantizer associated with the corresponding optimal distortion just obtained. It may be recalled in this respect that features (number of quanta, centroids and limit values) defining this quantizer are stored at the encoder in association with the distortion it generates, as already explained.

An example of convergence of the algorithm is shown on Figure 30. One clearly understands than the optimal non-saturation is not trivial because it does not correspond to the smallest encodable one. Actually, the smallest encodable non-saturation implies a lot of effort of encoding for the encoded DCT coefficient and leads to a big encoding rate. By encoding more coefficients, i.e. taking a set bigger than the smallest encodable one, one does not need to encode the coefficient too much and the rate is smaller. On the other hand, encoding all coefficients is far from being optimal as seen on the figure; little rate is used on too many coefficients and finally this leads to a big total rate. The optimal set of non-saturation is a non trivial balance between the number of encoded coefficients and the amount of encoding on each coefficient.

Once the distortion target $D_n$ (depending on the base mode) of each DCT coefficient has been determined by the above process, one chooses the best optimal quantizer associated to this distortion. For instance one may take, from the list of optimal quantizers corresponding to the associated parameter $\beta$ of the DCT channel model, the quantizer with the least rate among quantizers having distortion less or equal than the target distortion $D_n$.

Then, quantization is performed by the chosen (or selected) quantizers to obtain the quantized data $X_{DCT,Q}$ representing the DCT image. Practically, these data are symbols corresponding to the index of the quantum (or interval or Voronoi cell in 1D) in which the value of the concerned coefficient of $X_{DCT}$ falls in.

The entropy coding may be performed by any known coding technique like VLC coding or arithmetic coding. Context adaptive coding (CAVLC or CABAC) may also be used.

The encoded data can then be transmitted together with parameters allowing in particular the decoder to use the same quantizers as those selected and used for encoding as described above.

According to a first possible embodiment, the transmitted parameters may include the parameters defining the distribution for each DCT channel, i.e. the
parameter $\alpha$ (or equivalently the standard deviation $\sigma$) and the parameter $\beta$ computed at the encoder side for each DCT channel.

Based on these parameters received in the data stream, the decoder may deduce the quantizers to be used (a quantizer for each DCT channel) thanks to the selection process explained above at the encoder side (the only difference being that the parameters $\beta$ for instance are computed from the original data at the encoder side whereas they are received at the decoder side).

Dequantization (step 332 of Figure 4) can thus be performed with the selected quantizers (which are the same as those used at encoding because they are selected the same way).

According to a possible variation of this first embodiment, the parameters transmitted in the data stream include a parameter representative of the set $I^o$ of non-saturated coefficients which was determined at the encoder side to minimize the rate (i.e. the set for which the minimum rate $R_{mn}$ was obtained). In this variation, it is thus unnecessary to seek the relevant non-saturated coefficient set by optimisation and the process of selecting quantizers to be used is thus faster (part I of the process described above only).

According to a second possible embodiment, the transmitted parameters may include identifiers of the various quantizers used in the pool of quantizers (this pool being common to the encoder and the decoder) and the standard deviation $\sigma$ (or equivalently the parameter $\alpha$).

Dequantization (step 332 of Figure 4) can thus be performed at the decoder by use of the identified quantizers.

Further embodiment of decoder

The following description relates to a further embodiment of a decoder in accordance with the present invention.

Referring again to Figure 4, which illustrates the associated enhancement video decoder 33, it will be recalled that, from the received parameters 21, the channel models are reconstructed, meaning that a probabilistic distribution $\text{GGD}(c_{ij},P)$ is known for each encoded DCT coefficient of a channel $i$.

Quantizers are chosen 330 from the pool of quantizers, possibly based on these probabilistic distributions.

Next decoding, from the bit-stream, and dequantizing blocks of encoded quantized DCT coefficients of the first enhancement image are performed.
This comprises the following operations:
- an entropy decoder 331 is applied to the received enhancement layer bit-stream 20 to obtain the quantized DCT image $\overrightarrow{X}^{\text{DEC}}$. As suggested above, conventional Huffman codes can be used, possibly taking into account the probabilistic distributions; and
- a dequantization (or inverse quantization) 332 is then performed by using the chosen quantizers for each coefficient, to obtain a dequantized version of the DCT image. The dequantized version is referenced $\overrightarrow{X}^{\text{DEC}}_{Q^+}$, since it is different from the original version $X_{\text{DCT}}$ due to the lossy quantization.

The present embodiment particularly focuses on the rest of the decoding process, from the dequantized DCT coefficients $\overrightarrow{X}^{\text{DEC}}_{Q^+}$ of that dequantized image, as described now with reference to Figure 31.

As shown in this Figure, the decoding method according to the present embodiment comprises a step of merging 38 the dequantized DCT blocks $\overrightarrow{X}^{\text{DEC}}_{Q^+}$ of dequantized DCT coefficients with DCT residual blocks (or "predictor blocks") $Y$.

The DCT residual blocks $Y$ are generated by an enhancement prediction module 40 which is further detailed below.

The DCT blocks $\overrightarrow{X}^{\text{DEC}}_{Q^+}$ form a first version of the residual enhancement image currently decoded, while the DCT residual blocks $Y$ form, at least partly, a second version of the same residual enhancement image, that is temporally predicted based on base layer motion information and an already decoded UHD image of the video, as explained below.

The merger of the blocks $\overrightarrow{X}^{\text{DEC}}_{Q^+}$ with the blocks $Y$ may be a probabilistic merging process that is based on the parameters 21 (i.e. the probabilistic distributions of the DCT coefficients as determined by the encoder) and on a second probabilistic distribution that characterizes the temporal prediction of the enhancement layer by the module 40. In particular, the second probabilistic distribution is a probabilistic distribution of the differences between the coefficients of the DCT residual blocks $Y$ and the dequantized DCT coefficients of the dequantized DCT blocks $\overrightarrow{X}^{\text{DEC}}_{Q^+}$.

Figure 32 illustrates the generation of the DCT residual blocks $Y$, i.e. of transformed residual blocks of the enhancement image associated with a current image $I$ to decode.
This prediction successively consists in temporally predicting current enhancement image in the pixel domain (thanks to up-sampled motion information), computing the pixel difference data between temporal predicted image and up-sampled reconstructed base image and then applying a DCT transform on the difference image.

It is first assumed that the image $I^B$ of the base layer corresponding to the current image $I$ to decode has already been decoded by the base layer decoder using temporal prediction based on an already-decoded reference image $I_{nB}$ of the base layer. For each block $B$ or macroblock $MB$ of the image $I^B$ (depending on the granularity of the prediction), motion information $MI$ is known and temporarily stored when decoding the base layer, for the needs of the present invention.

This motion information comprises, for each block or macroblock, a base motion field $BMF$ (including a motion vector and a reference image index) and a base residual $BR$, as well-known by one skilled in the art of video coding.

Each of the base blocks or base macroblocks that have been temporally predicted in the base layer are now successively considered.

It may be noted that the present invention also applies when not all the images $I^B$ of the base layer are encoded in the bit-stream $EBS$. For example, an image $I^B$ of the base layer may be obtained by interpolating other decoded base images. In that case, the available motion information for those other decoded base images may also be interpolated to provide motion information specific to blocks or macroblocks of the interpolated base image $I^B$. The following explanation also applies for such kind of base image $I^B$.

First, the corresponding motion information is up-sampled 400 into high resolution corresponding to the resolution of the enhancement layer (e.g. UHD). It is shown in the Figure by the references $UMF$ (up-sampled motion field) and $UR$ (up-sampled residual).

In particular, this up-sampling comprises for each base macroblock:

- the up-sampling of the macroblock partitioning (into blocks) to the resolution level of the enhancement layer. In the above example of a dyadic spatial scalability (UHD v.s. HD), up-sampling the partition consists in multiplying the width and height of macroblock partitions by a factor of 2;
- the up-sampling, by a factor of 2 (in width and height), of the base residual associated with the base macroblock. This texture up-sampling process may use an interpolation filter that is identical to that used in inter-layer residual prediction mechanisms of the SVC scalable video compression standard; and
- the up-sampling, by two (x- and y- coordinates), of the motion vector associated with the base macroblock.

Once the up-sampling 400 has been performed, the generation of a DCT residual macroblock Y comprises a motion compensated prediction step 405 from the decoded UHD image that temporally corresponds to the reference base image $I_R^B$ used for the decoding of the base layer, and based on the up-sampled motion information UMF and UR. That decoded UHD image is considered, for the temporal prediction, as the reference decoded image $I_{R}^{UHD}$.

It may for example be the reconstructed UHD image that temporally precedes the current image to decode, as shown in the Figure.

This motion compensation 405, in the pixel domain, leads to obtaining, using the motion information, motion predictor blocks from the decoded reference high resolution image $I_R^{UHD}$. In particular, the up-sampled prediction information is applied to the reference decoded image $I_{R}^{UHD}$ to determine predicted macroblocks.

One may note that the motion compensation results in a partially-reconstructed image. This is because the macroblocks reconstructed by prediction are obtained at spatial positions corresponding to INTER macroblocks in the base image $I_B$ only (because there is no motion information for other macroblocks). In other words, there is no predicted block that is generated for the macroblocks collocated with INTRA macroblocks in the base layer.

Next, residual blocks are obtained by subtracting 410 each motion predictor block from a corresponding (i.e. collocated) up-sampled block in the up-sampled decoded base image (which is obtained by the up-sampling 32 of Figure 2). This step calculates the difference image (or residual) between the temporally predicted image and the up-sampled reconstructed base layer image. This difference image has the same nature as the residual enhancement image.

The module 40 ends by applying 415 a block-based transform, e.g. DCT on 8x8 blocks, on the obtained residual blocks to obtain transformed residual blocks that are the DCT residuals Y discussed above.

Therefore, a plurality of DCT residual macroblocks Y is obtained for the current image to decode, which generally represent a partial predicted enhancement image.

The next steps of the decoding method according to the invention may be applied to the entirety of that plurality of macroblocks Y, or to a part of it depending for example on the base coding mode (P image Inter prediction, B image Inter prediction,
Skip mode) in which case only the DCT predictor macroblocks Y and the dequantized DCT macroblocks $\overline{X}_{Q^4}^{\text{DEC}}$ collocated with base macroblocks having the same coding mode are handled together.

For the rest of the description, macroblocks Y collocated with P, B and SKIP base macroblocks are considered separately, as was done at the encoder when determining the probabilistic distribution of each DCT channel.

Based on the considered DCT residual macroblocks Y, a probabilistic distribution of the differences between the coefficients in the transformed residual blocks Y and the dequantized transformed coefficients $\overline{X}_{Q^4}^{\text{DEC}}$ is calculated. This aims to model the noise associated with the motion prediction 405.

A probabilistic distribution may be obtained for the entire set of coefficients of the considered DCT residual macroblocks, or for each DCT channel / in which case the explanation below should be applied for the DCT coefficients of the same channel.

Each DCT residual macroblock Y made of DCT coefficients for the current image to decode is considered as a version of the original DCT coefficients that would have been altered through a communication channel. It has been observed that the quantity $Y - X_{\text{DCT}}$ (i.e. the noise of the residual Y compared to the DCT coefficients before encoding) can be well modelled by a generalized Gaussian distribution as introduced above: DCT $(Y - X_{\text{DCT}}) \approx \text{GGD}(\alpha_N, \beta_N)$

By knowing the statistical distribution of the predictor noise $(Y - X_{\text{DCT}})$, it is therefore possible to retrieve a good approximation of the original DCT coefficients of blocks $X_{\text{DCT}}$.

Since the exact coefficients of DCT image $X_{\text{DCT}}$ are not known by the decoder (because of the quantization 192 at the encoding side), the exact prediction noise cannot be modelled. However, the inventors have observed that using the dequantized DCT macroblocks $\overline{X}_{Q^4}^{\text{DEC}}$ instead of the original DCT macroblocks $X_{\text{DCT}}$ provides a GGD modelling that is close to the theoretical modelling with $X_{\text{DCT}}$.

For this, the modelling of the predictor noise thus comprises fitting a Generalized Gaussian Distribution model onto the differences between the coefficients in the transformed residual blocks Y and the dequantized transformed coefficients $\overline{X}_{Q^4}^{\text{DEC}}$. 
The same mechanisms based on the first and second moments as described above can be applied to obtain the two parameters $\alpha_m, \beta_m$ (either for all the considered macroblocks $Y$ and $X_{DCT}^{DEC, j}$, or for each DCT channel of coefficients in those macroblocks).

Next, the merging 38 of the considered macroblocks $Y$ and $X_{DCT}^{DEC, j}$ is performed, in the DCT domain.

As introduced above, it is based on the probabilistic distribution of the DCT coefficient of $X_{DCT}$ that is obtained through the parameters $\alpha_i, \beta_i$ of each DCT channel: $P(X_i=x)=\text{GGD}(\alpha_i, \beta_i, x)$, and on the probabilistic distribution of the predictor noise as determined above (parameters $\alpha_m, \beta_m$ possibly for the considered DCT channel): $P(Z=z)=\text{GGD}(c\alpha_N, \beta_N, x)$ where $Z$ is the estimated noise $Y-X^N_{DCT}$.

For each coefficient $i$ of the considered macroblocks, the merged value can take the form of a probabilistic estimation of the original DCT coefficients value, given the known quantization interval of this DCT coefficient, and an aside approximation of the coefficient resulting from its motion compensated temporal prediction (blocks $Y$).

For example, a merged value according to the invention, denoted $\tilde{x}_j$, may be the expectation (or the "expected value") of the considered coefficient given the quantization interval $Q_m$ associated with the value of the corresponding quantized transformed coefficient in $\tilde{x}_{DCT}$ and given its corresponding value $Y_0$ in the residual blocks $Y$. The quantization interval $Q_m$ is directly retrieved from the quantized DCT coefficient obtained from the bit-stream 20, since its value $a_m$ is the index of the quantization interval $Q_m$ given the quantizer used.

Such expectation is calculated based on the probabilistic distributions mentioned previously, for example

$$\tilde{x}_j = \frac{\int x \text{GGD}(\alpha_i, \beta_i, x) \text{GGD}(\alpha_N, \beta_N, x - Y_0) dx}{\int \text{GGD}(\alpha_i, \beta_i, x) \text{GGD}(\alpha_N, \beta_N, x - Y_0) dx}.$$

The probabilistic calculation of $\tilde{x}_j$ is illustrated in Figure 33.

In this Figure, the GGD distribution of $X_{DCT}$ as well as the statistical distribution of the prediction noise $X_{DCT}-Y_0$ have been drawn. The quantization interval $Q_m$ associated with the considered DCT coefficient $i$ is also indicated.
The two distributions are multiplied over the interval $Q_m$ to calculate the desired conditional expected value of $X_{DCT}$. The integrals of those distributions can be computed using Riemann sums over the quantization interval.

Thanks to the invention, there is no need to force the obtained value $Y_0$ to be within the current quantization interval $Q_m$. On the contrary, the fact that it could be outside that interval (as shown in the Figure) is thus taking into account to refine the DCT decoded coefficients.

The values $\hat{X}_i$ calculated for the DCT coefficients of all the considered macroblocks are stored in memory to form, at least partially, the merged enhancement image corresponding to the current image to decode.

Since no fitting and merging step is applied to INTRA macroblocks (because there is no motion information), their values obtained in the dequantized DCT macroblocks $\hat{X}_{i}^{DEC}$ may be used to form the remaining part of the merged enhancement image. Finally, an entire merged enhancement image is obtained which is input to the inverse transform IDCT 34 as shown in Figures 2 and 31.

The present embodiment has been illustrated and provides significant improvements in rate-distortion, of about several dBs. As explained in detail, the present embodiment relies on:

- statistically modelling, at the encoder, the DCT coefficients of the residual enhancement image and then transmitting the obtained model parameters to the decoder;
- statistically modelling, at the decoder, the DCT coefficients of a predicted residual enhancement image (issued from temporal prediction using the motion information of the base layer); and
- probabilistically estimating (through a conditional expectation calculation) the combined DCT coefficients of these two residual enhancement images, based on their statistical modelling.

Figure 33 illustrates the performance of the present embodiment, in which the rate-distortion curves are plotted when the merging according to the present embodiment is respectively not implemented and implemented.

The Figure shows that an improvement the codec rate distortion performance is obtained, especially at low bitrates. This may be understood intuitively, since the quantization intervals get larger as the bitrate decreases, therefore increasing...
the relevant information brought by the temporal DCT residuals $Y$ compared to the quantization level obtained by the dequantization step 332.

One may then note that the present embodiment also works for zero bitrate (meaning that no enhancement layer bitstream 20 is encoded or received by the decoder). In that case the parameters $21 \ (\alpha, \beta, \text{for each DCT channel})$ are received and are used with the parameters $\alpha_N, \beta_N$ calculated with the present embodiment to obtain an improvement of the decoding quality of the base layer by several dBs.

Furthermore, the above performance is obtained with no complexity cost at the encoding side and with no additional bitrate when the parameters $21$ are already needed and transmitted (e.g. for selecting the quantizers and/or entropy decoding). The complexity increase due to the merging step remains reasonable at the decoding side.

The above examples are merely embodiments of the invention, which is not limited thereby.
CLAIMS

1. A method of encoding video data comprising:
   encoding video data having a first resolution in conformity with HEVC to obtain video data of a base layer; and
   decoding the base-layer video data in conformity with HEVC, upsampling the decoded base-layer video data to generate decoded video data having a second resolution higher than said first resolution, forming a difference between the generated decoded video data having said second resolution and further video data, having said second resolution and corresponding to said first-resolution video data, to generate data of a residual image, and compressing the residual-image data to generate video data of an enhancement layer.

2. A method as claimed in claim 1, wherein the compression of the residual-image data does not involve temporal prediction.

3. A method as claimed in claim 1 or 2, wherein the compression of the residual-image data does not involve spatial prediction.

4. A method as claimed in claim 1, 2 or 3, wherein the compression of the residual-image data comprises applying a discrete cosine transformation (DCT) to obtain DCT coefficients.

5. A method as claimed in claim 4, further comprising employing a parametric probabilistic model of the DCT coefficients.

6. A method as claimed in claim 5, wherein such a parametric probabilistic model is obtained for each type of DCT coefficient.

7. A method as claimed in claim 5, wherein the DCT is a block-based DCT and such a parametric probabilistic model is obtained for each DCT coefficient position within a DCT block.

8. A method as claimed in claim 7, further comprising fitting the parametric probabilistic model onto respective collocated DCT coefficients of some or all DCT blocks of the residual image.

9. A method as claimed in any one of claims 5 to 8, wherein the compression of the residual-image data comprises employing the parametric probabilistic model to choose quantizers from a pool of available quantizers and quantizing the residual data using the chosen quantizers.
10. A method as claimed in claim 9, wherein the available quantizers of said pool are pre-computed quantizers dedicated to a parametric probabilistic model.

11. A method as claimed in any one of claims 5 to 10, wherein the compression of the residual-image data comprises entropy encoding of quantized symbols and employing a parametric probabilistic model to obtain a probabilistic distribution of possible symbols of an alphabet associated with each DCT coefficient, which alphabet is used for the entropy encoding.

12. A method as claimed in claim 11 when read as appended to claim 9 or 10, wherein the same parametric probabilistic model is employed both for choosing quantizers and for entropy encoding.

13. A method as claimed in any one of claims 5 to 12, wherein the parametric probabilistic model is a Generalised Gaussian Distribution GGD(a, β) having a zero mean.

14. A method as claimed in any one of claims 5 to 13, wherein parameters of the parametric probabilistic model are determined in dependence upon one or more of:
   - a video content;
   - an index of the DCT coefficient within a DCT block;
   - an encoding mode used for a collocated block of the base layer; and
   - a size of block to encode.

15. A method as claimed in any one of claims 5 to 14, wherein information about parameters of the parametric probabilistic model is supplied to a decoder.

16. A method as claimed in any one of claims 5 to 14, wherein information about a set of saturated coefficients determined at the encoder to minimize a rate is supplied to a decoder.

17. A method as claimed in claim 9 or 10, wherein information about the chosen quantizers is supplied to a decoder.

18. A method as claimed in claim 9 or 10, wherein the quantizers are chosen based on a rate-distortion criterion.

19. A method of decoding a scalable bitstream, comprising:
   decoding, in conformity with HEVC, encoded video data of a base layer of the bitstream to obtain video data having a first resolution, and upsampling the first-resolution video data to generate video data having a second resolution higher than said first resolution; and
decoding compressed video data of an enhancement layer of the bitstream to obtain data of a residual image, and forming a sum of the generated second-resolution video data and the residual-image data to generate decoded video data having said second resolution.

20. A method as claimed in claim 19, wherein the decoding of the compressed video data of the enhancement layer does not involve temporal prediction.

21. A method as claimed in claim 19 or 20, wherein the decoding of the compressed video data of the enhancement layer does not involve spatial prediction.

22. A method as claimed in claim 19, 20 or 21, wherein the compressed video data of the enhancement layer comprises encoded discrete cosine transformed (DCT) coefficients.

23. A method as claimed in claim 22, further comprising employing a parametric probabilistic model of the DCT coefficients.

24. A method as claimed in claim 23, wherein such a parametric probabilistic model is obtained for each type of DCT coefficient.

25. A method as claimed in claim 23, wherein the DCT is a block-based DCT and such a parametric probabilistic model is obtained for each DCT coefficient position within a DCT block.

26. A method as claimed in any one of claims 23 to 25, wherein the decoding of the compressed video data of the enhancement layer comprises employing the parametric probabilistic model to choose quantizers from a pool of available quantizers and using the chosen quantizers for inverse quantization of the encoded DCT coefficients.

27. A method as claimed in claim 26, wherein the available quantizers of said pool are pre-computed quantizers dedicated to a parametric probabilistic model.

28. A method as claimed in any one of claims 22 to 27, wherein the decoding of the compressed video data of the enhancement layer comprises entropy decoding of encoded quantized symbols obtained from the compressed video data to generate quantized DCT coefficients and employing a parametric probabilistic model to obtain a probabilistic distribution of possible symbols of an alphabet associated with each DCT coefficient, which alphabet is used for the entropy decoding.
29. A method as claimed in claim 28 when read as appended to claim 26 or 27, wherein the same parametric probabilistic model is employed both for choosing quantizers and for entropy decoding.

30. A method as claimed in any one of claims 23 to 29, wherein the parametric probabilistic model is a Generalised Gaussian Distribution $\text{GGD}(a, \beta)$ having a zero mean.

31. A method as claimed in any one of claims 23 to 30, wherein information about parameters of the parametric probabilistic model is received from an encoder.

32. A method as claimed in any one of claims 23 to 30, wherein information about a set of saturated coefficients determined at an encoder to minimize a rate is received from the encoder.

33. A method as claimed in claim 26 or 27, wherein information about the chosen quantizers is received from an encoder.

34. A method as claimed in claim 26 or 27, wherein the quantizers are chosen based on a rate-distortion criterion.

35. Apparatus for encoding video data comprising:

- means for encoding video data having a first resolution in conformity with HEVC to obtain video data of a base layer;
- means for decoding the base-layer video data in conformity with HEVC;
- means for upsampling the decoded base-layer video data to generate decoded video data having a second resolution higher than said first resolution;
- means for forming a difference between the generated decoded video data having said second resolution and further video data, having said second resolution and corresponding to said first-resolution video data, to generate data of a residual image; and
- means for compressing the residual-image data to generate video data of an enhancement layer.

36. Apparatus for decoding a scalable bitstream, comprising:

- means for decoding, in conformity with HEVC, encoded video data of a base layer of the bitstream to obtain video data having a first resolution;
- means for upsampling the first-resolution video data to generate video data having a second resolution higher than said first resolution;
- means for decoding compressed video data of an enhancement layer of the bitstream to obtain data of a residual image; and
means for forming a sum of the generated second-resolution video data and the residual-image data to generate decoded video data having said second resolution.

37. A program which, when executed by a computer or processor, causes the computer or process to carry out a method of encoding as claimed in any one of claims 1 to 18.

38. A program which, when executed by a computer or processor, causes the computer or process to carry out a method of decoding as claimed in any one of claims 19 to 34.
Fig. 3
Fig. 5

Fig. 6
Fig. 7

Distribution of DCT coefficients for a channel

Minimum DCT value

Maximum DCT value

$t_m$

$t_{m+1}$

$C_m$
Distribution of DCT coefficients for a channel

-2 -1 0 1 2 3 4

symbols $a_m \Rightarrow$ alphabet $A = \{a_m\}$

$P_0 \ P_1 \ P_2 \ P_3$

probability of occurrence

Fig. 8
Alphabets $A_i$ of the quantized DC T symbols and associated probabilistic distributions \{$P_{i,m}$\}_m

Discard

Intra-block grouping

Inter-block grouping

Resulting alphabets and associated probability distributions

Fig. 10
Group $A_{\min} \& A_{\max} \rightarrow$ alphabet $A_{\min} \times A_{\max}$
- Product set of symbols:
  $\{c\} = \{(a, b), a \in A_{\min}, b \in A_{\max}\}$
- Update probabilities
- Append lists of concerned DCT coeffs

Search for $A_{\min}$ with minimum entropy: $H(A_{\min}) = -\sum p_{\min} \log_2(p_{\min})$

Search for $A_{\max}$ with maximum entropy:
- $H(A_{\max}) = \max\{H(A_k), k \neq i\}$
- $\text{Card}(A_{\min} \times A_{\max}) = \text{Card}(A_{\min}) \times \text{Card}(A_{\max}) < \text{MAX\_ALPHABET\_SIZE}$

One pair found? yes  no

Fig. 13

END
Obtain symbol group associated with macroblock mode coding (UP or SKIP) and restricted alphabets indicated by the skip flags.

S ← set of symbol groups in S

Encode flags associated with current macroblock.

g ← first group of symbols in S

block_index ← 0

Obtain symbol value for group g:

symbolVal[g] ← symbolVal[g] + quantizedcoeffs[g];

for each g = 1, ..., gnbDCTcoeffs

Obtain Huffman code associated with symbolVal[g].

Write Huffman code to output bitstream.

block_index ← block_index + g.nbBlock

block_index < 4 (nb block per macroblock)?

Yes

block_index ← block_index + g.nbBlock

Obtain last group of symbols in S?

Yes

End

No

g ← next group in S

1535

1534

1533

1532

1531

1530

1529

1528

1527

1526

1525

1524

1523

1522

1521

1520

1519

1518

1517

1516

1515

1514

1513

1512

1511

1510

1509

1508

1507

1506

1505

1504

1503

1502

1501

1500

Fig. 15
Obtain alphabet $F'$ according to base macroblock coding mode

Decode macroblock flag $f_{mb}$ associated with current macroblock

$S$ ← set of symbol group associated with macroblock coding mode (IP or SKIP) and restricted alphabets indicated by the decoded skip flags

$g$ ← first group of symbols in $S$

Decoded binary code according to Huffman tree associated with group $g$

Block_index $\rightarrow$ block_index $+ g.nhBlock$

Obtain DCT coefficients values linked to obtained Huffman tree leaf

Block_index $< 4$ (nb block per macroblock)

$g$ ← next group of symbols in $S$

End
Fig. 17

The invention

Entropy

Conventionnel Huffman coding

Fig. 18

Granularity level

Extractible subregion

entry points

Residual enhancement image

Macroblocks
One self-contained coded subregion

Fig. 19
Alphabets \( \mathcal{A}_i \) of quantized DCT coefficients and associated \( \{ p_{m,n} \} \) and \( \{ p_{MB,m,n} \} \)

Compute \( P_{B,m} \) with associated \( \{ P_{B,m} \} \)

Associate flag \( f_i \) with each block

Determine \( a, b \) to restrict \( F \)

Restrict alphabets \( \mathcal{A}_i \) given flags \( f_i \)

Restricted alphabets \( \mathcal{A}_i \) and normalized probabilities \( \{ P_{F,m} \} \)

Fig. 21
Total rate (bit/block)

Number of coded DCT coefficients = Cardinal of encodable non-saturation sets

- 30 iterations
- 11 iterations
- 1 iteration

Optimal saturation with associated optimal rate
\[ P(Z = z) = \text{GGD} \left( \alpha_x, \beta_x, z \right) \]

\[ z = x - Y_0 \]

\[ P(X_i = x) = \text{GGD} \left( \alpha_i, \beta_i, x \right) \]

Fig. 33
A. CLASSIFICATION OF SUBJECT MATTER

INV. H04N7/26  H04N7/50

ADD.

According to International Patent Classification (IPC) also both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal , COMPENDEX, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>wO 2005/057933 Al (KONINKL PHI LIPS ELECTRONICS NV) [NL]; VAn VUGHT HENRICUS A G [NL]; BRULS) 23 June 2005 (2005-06-23) abstract; figure 1 page 1, line 13 - page 2, line 21</td>
<td>1-5 , 11, 19-23 , 28,35-38</td>
</tr>
<tr>
<td>A</td>
<td>BRUNO MACCHIAVELLO ET AL: &quot;A STATISTICAL MODEL FOR A MIXED RESOLUTION WYNER-ZIV FRAMEWORK&quot;, 26. PICTURE CODING SYMPOSIUM; LISBON, 7 November 2007 (2007-11-07), XP030080372, abstract; figures 1, 2 sections 1-5</td>
<td>1-38</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the public date of another invention

"O" document relating to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

*I* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

+Y+ document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

+S+ document member of the same patent family

Date of the actual completion of the international search

4 September 2012

Date of mailing of the international search report

13/09/2012

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk

Tel. (+31-3170) 340-2040, Fax: (+31-3170) 340-3016

Authorized officer

Strei ch, Sebasti an
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 6 700 933 BI (WU FENG [CN] ET AL) 2 March 2004 (2004-03-02) abstract column 16, line 4 - column 17, line 50</td>
<td>1-38</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>WO 2005057933 AI</td>
<td>23-06-2005</td>
<td>CN 1890980 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1695555 AI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2007514359 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20060126984 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2007160300 AI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2005057933 AI</td>
</tr>
<tr>
<td>US 6700933 BI</td>
<td>02-03-2004</td>
<td>AT 291329 T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU 2959501 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 60109423 DI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 60109423 T2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1258147 AI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1511324 AI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 4842486 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2003523667 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TW 519843 B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 6700933 BI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 0162010 AI</td>
</tr>
</tbody>
</table>