

(21) Application No: 1100224.3
 (22) Date of Filing: 07.01.2011

(71) Applicant(s):
Canon Kabushiki Kaisha
 (Incorporated in Japan)
 30-2 Shimomaruko 3-chome, Ohta-Ku 146-8501,
 Tokyo, Japan

(72) Inventor(s):
Eric Nassor
Hervé Le Floch

(74) Agent and/or Address for Service:
Canon Europe Limited
 Patent Administration Dept, 3 The Square,
 Stockley Park, Uxbridge, UB11 1ET, United Kingdom

(51) INT CL:
H04N 7/68 (2006.01)

(56) Documents Cited:
WO 2009/057956 A1
Rane, S. et al., "Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting - a Wyner-Ziv Coding Approach", 2004, International Conference on Image Processing (IEEE), vol. 5, pages 3101-3104

(58) Field of Search:
 INT CL **H04N**
 Other: **Online: WPI, EPODOC, Inspec**

(54) Title of the Invention: **Improved reconstruction of at least one missing area of a sequence of digital images**
 Abstract Title: **Improved reconstruction of at least one missing area of a sequence of digital images**

(57) An encoded digital signal, comprising an image represented by a plurality of samples, is decoded. When a part of the encoded image is missing, a reconstruction is applied to form a reconstructed image. The reconstruction comprises setting a missing sample to a predicted value and obtaining a value of an estimated error between the predicted value and the corresponding missing value using additional data derived by the encoder from at least part of the encoded image and usable during decoding to correct the encoded image. A confidence evaluation of the estimated error is computed (355) using the estimated error values obtained for a selected plurality of predicted sample values. The confidence evaluation of the estimated error obtained is then used to improve the reconstruction of the encoded digital image to decode. Advantageously, the confidence evaluation of the estimated error can be computed adaptively and a better reconstruction of missing areas is obtained.

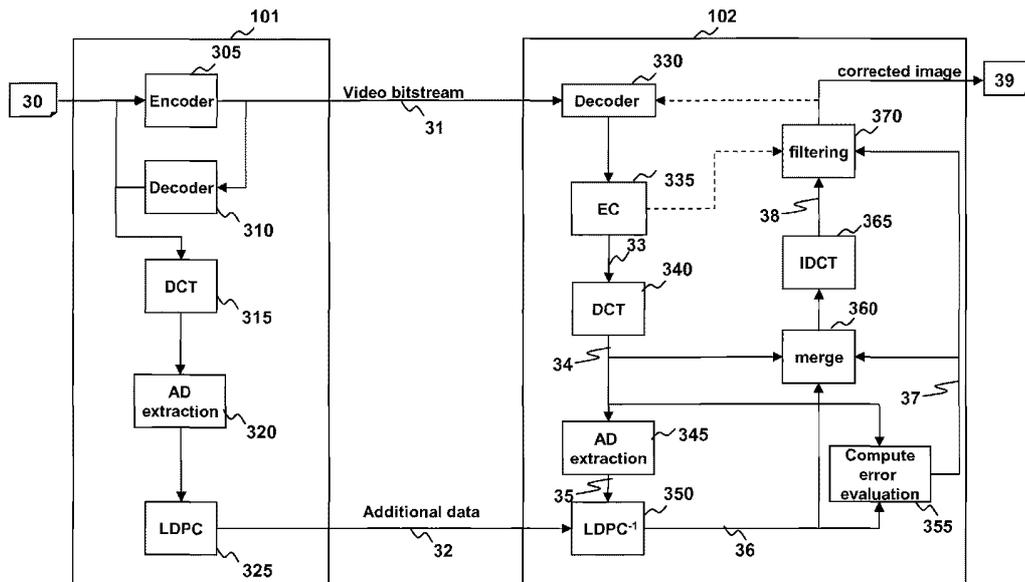


Fig. 3

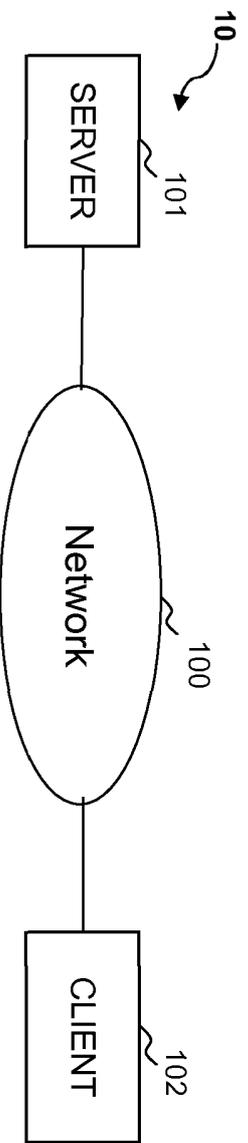


Fig. 1

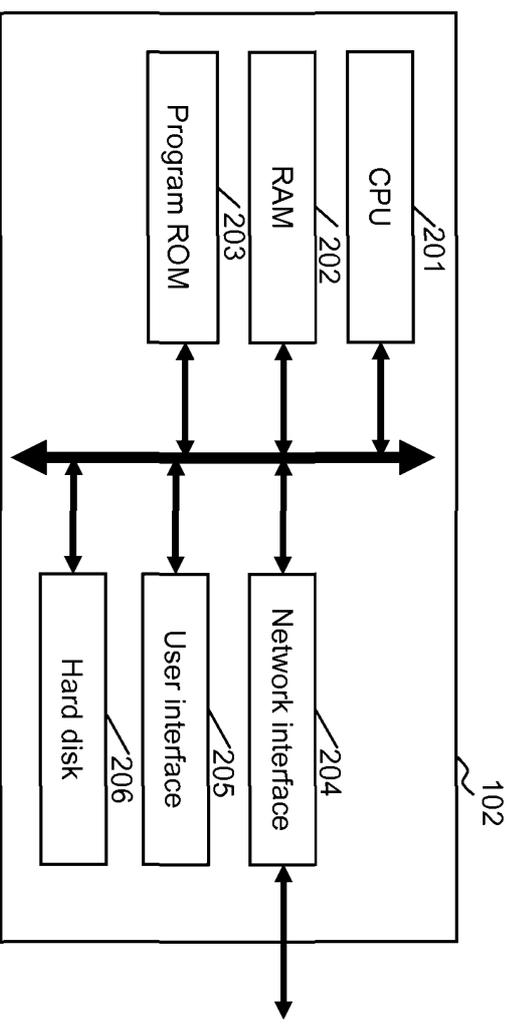


Fig. 2

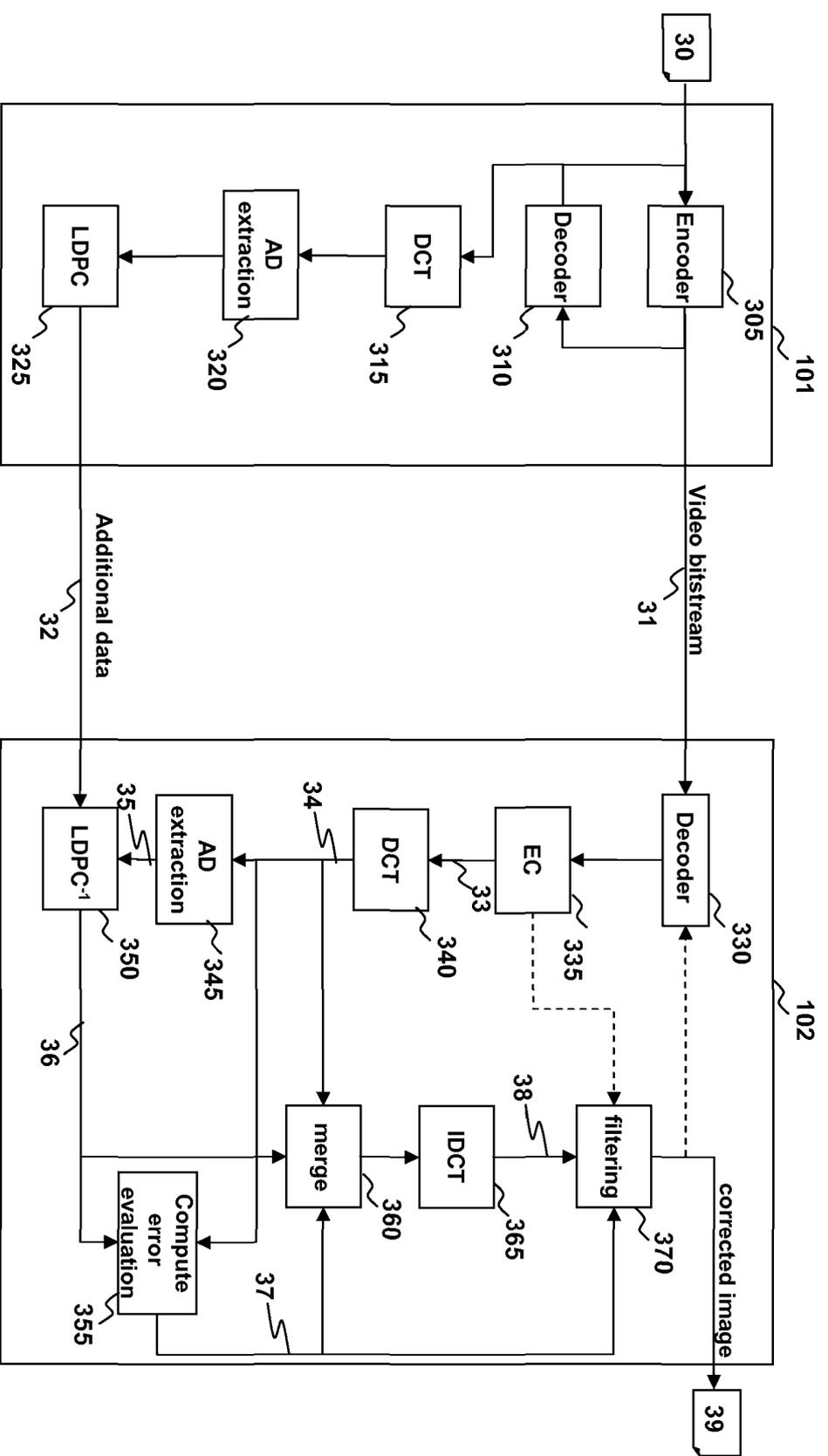


Fig. 3

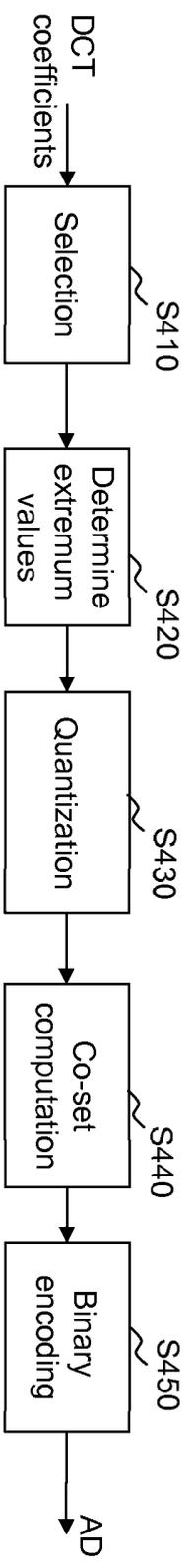


Fig. 4

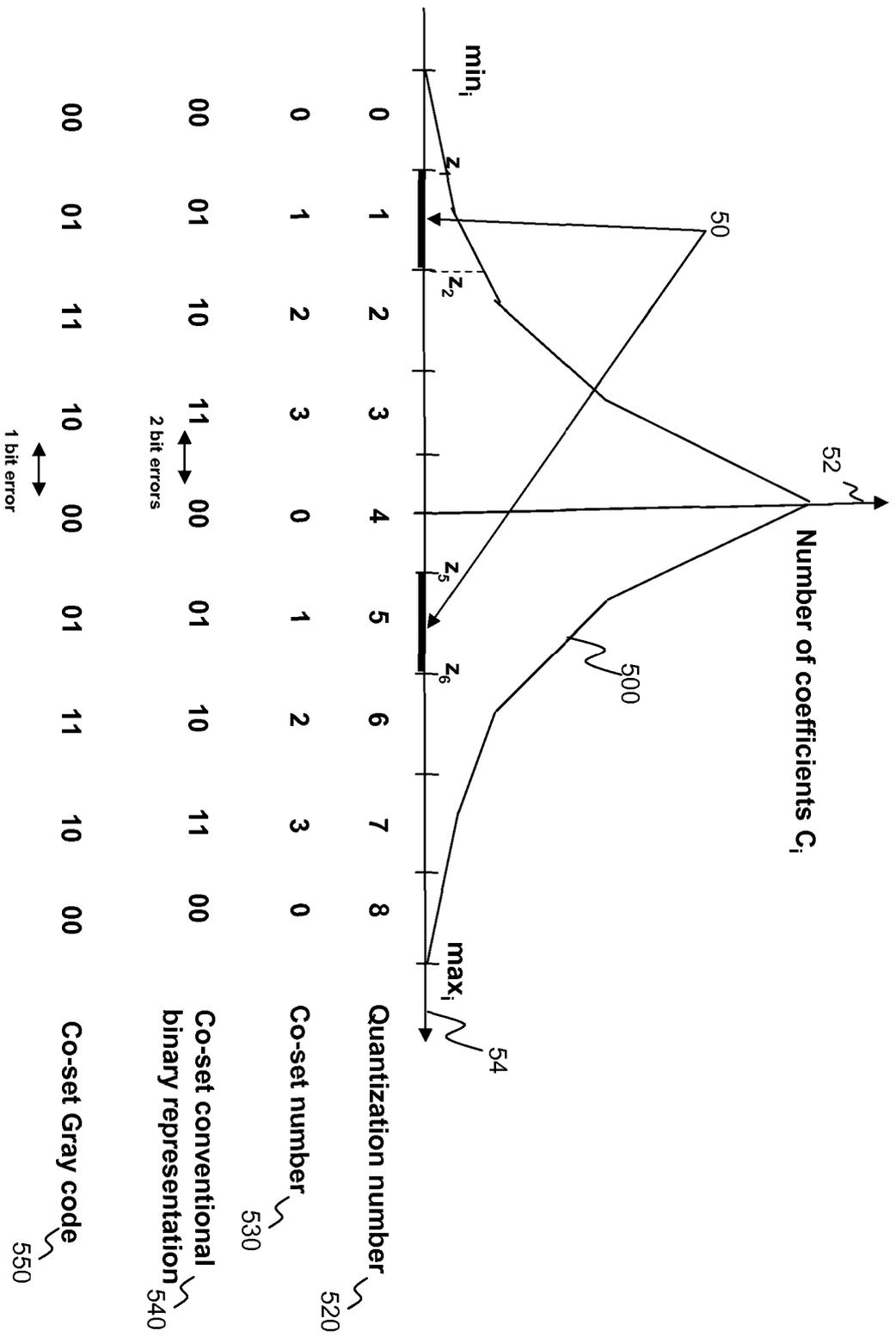


Fig. 5

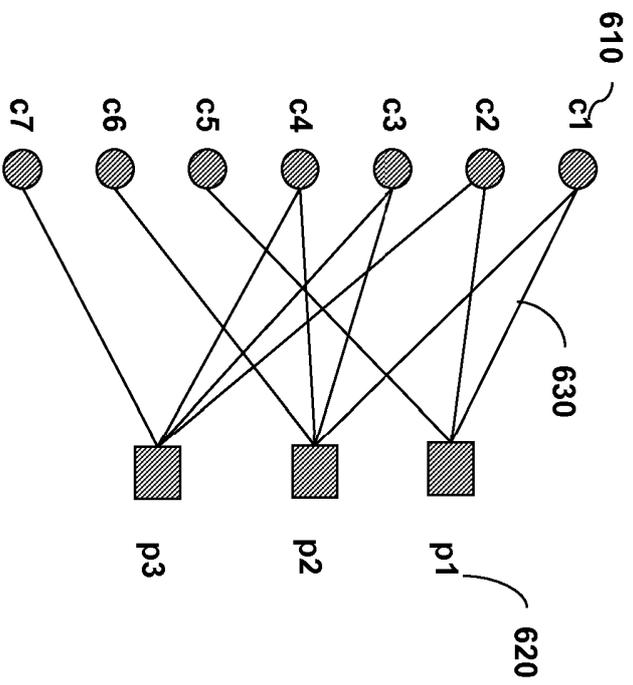


Fig. 6

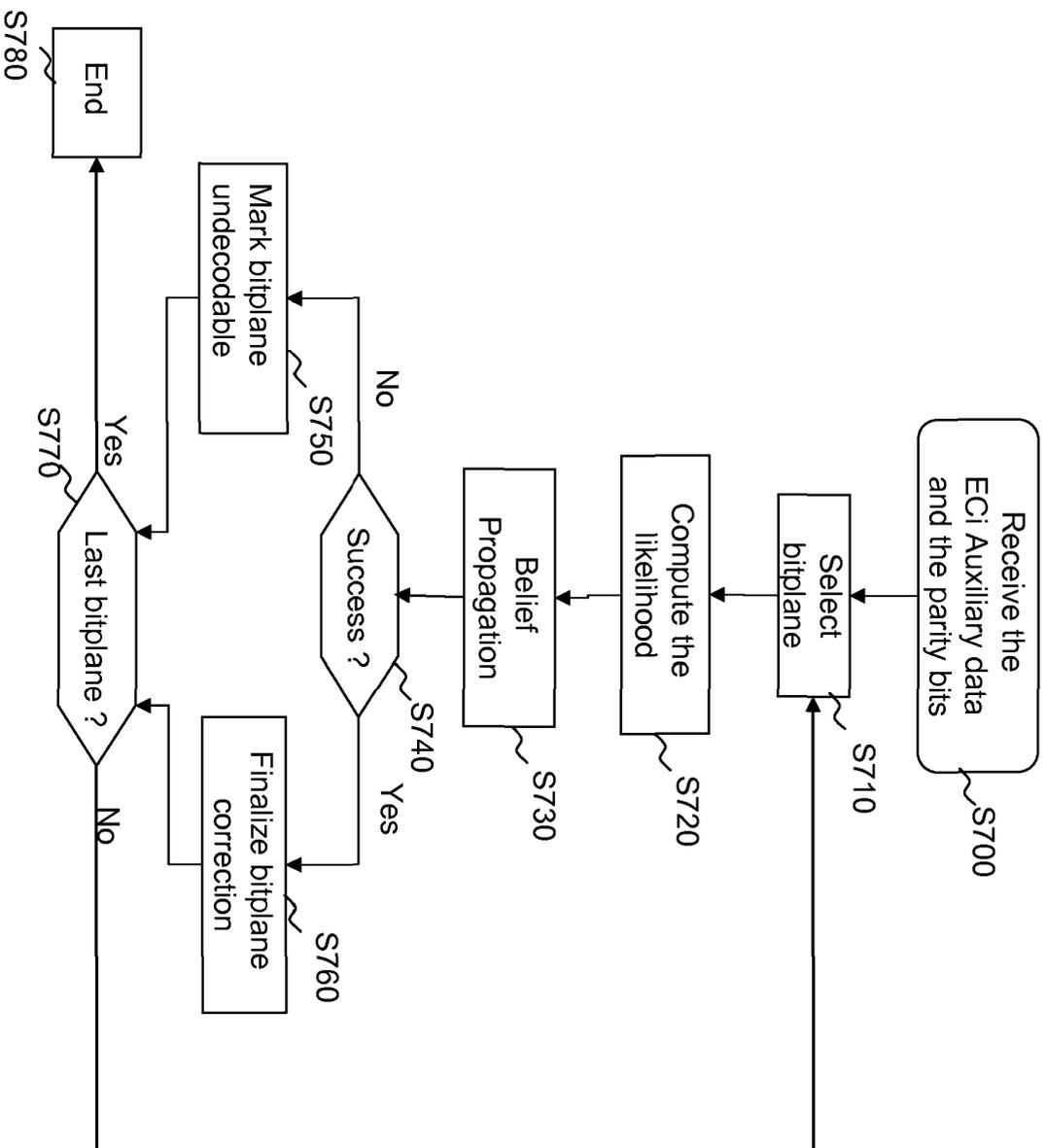


Fig. 7

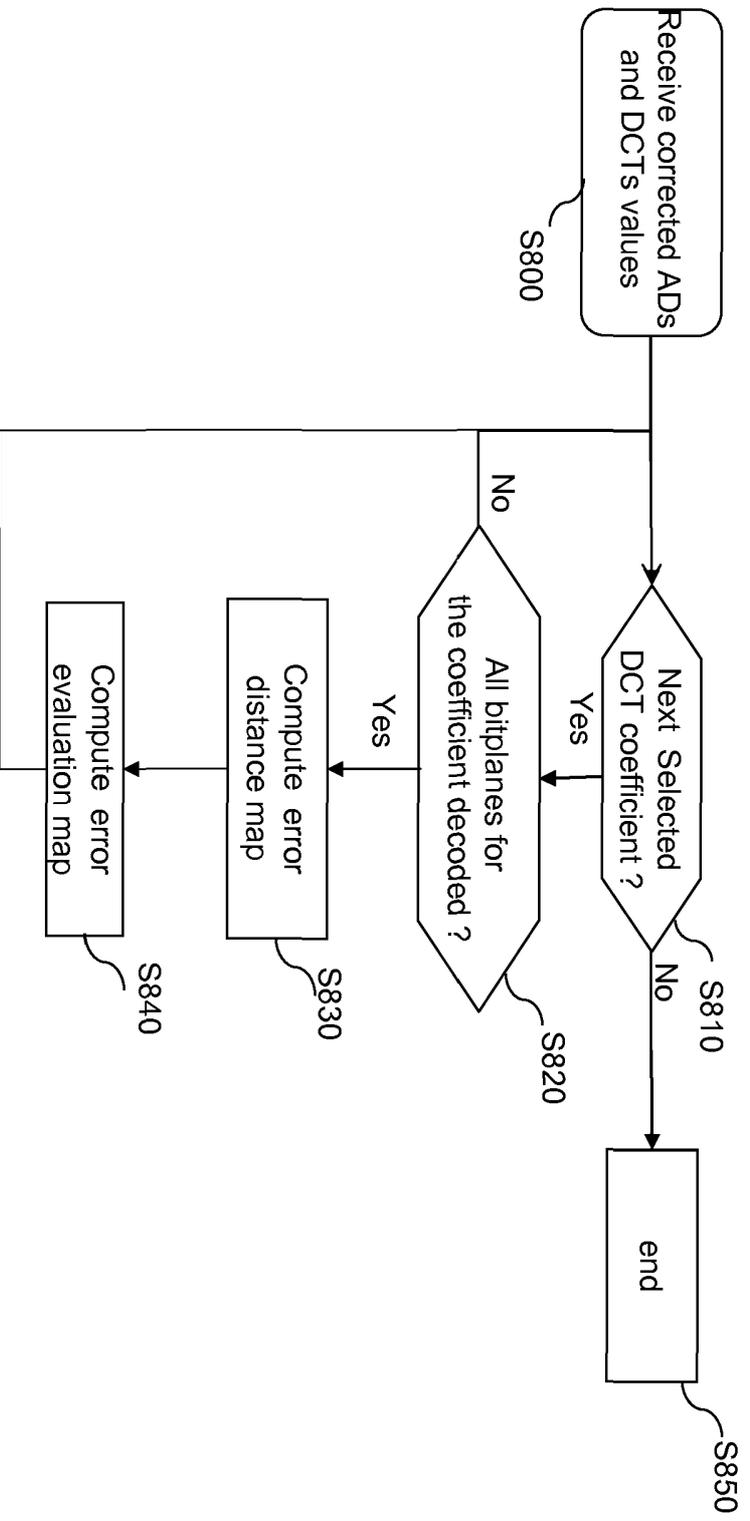


Fig. 8

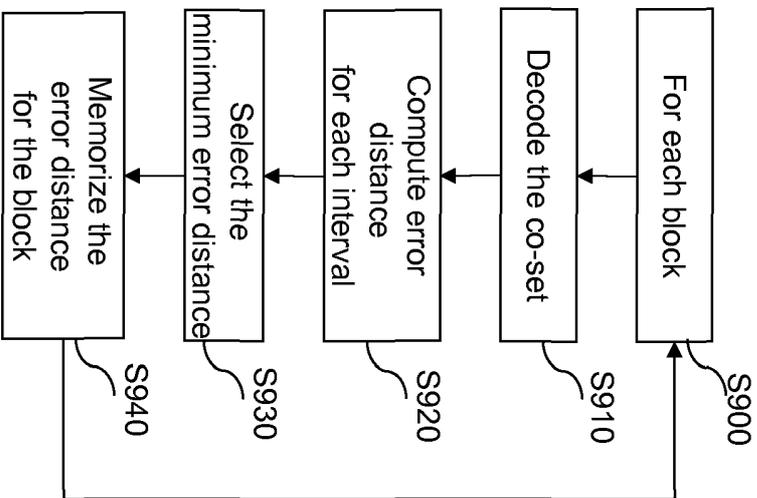


Fig. 9

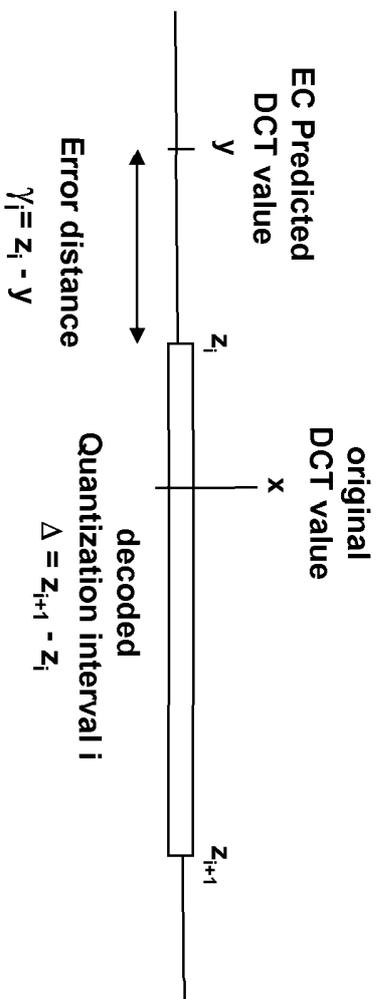


Fig. 10

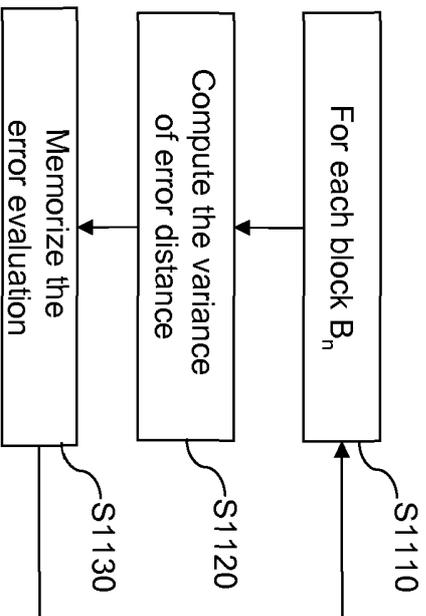


Fig. 11

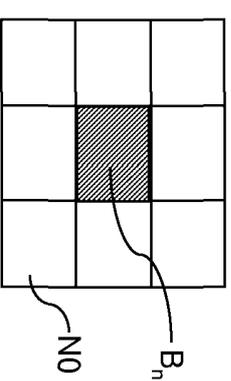


Fig. 12 a

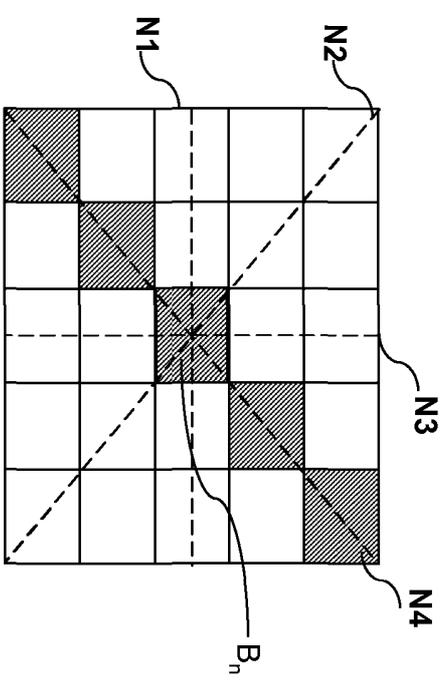


Fig. 12 b

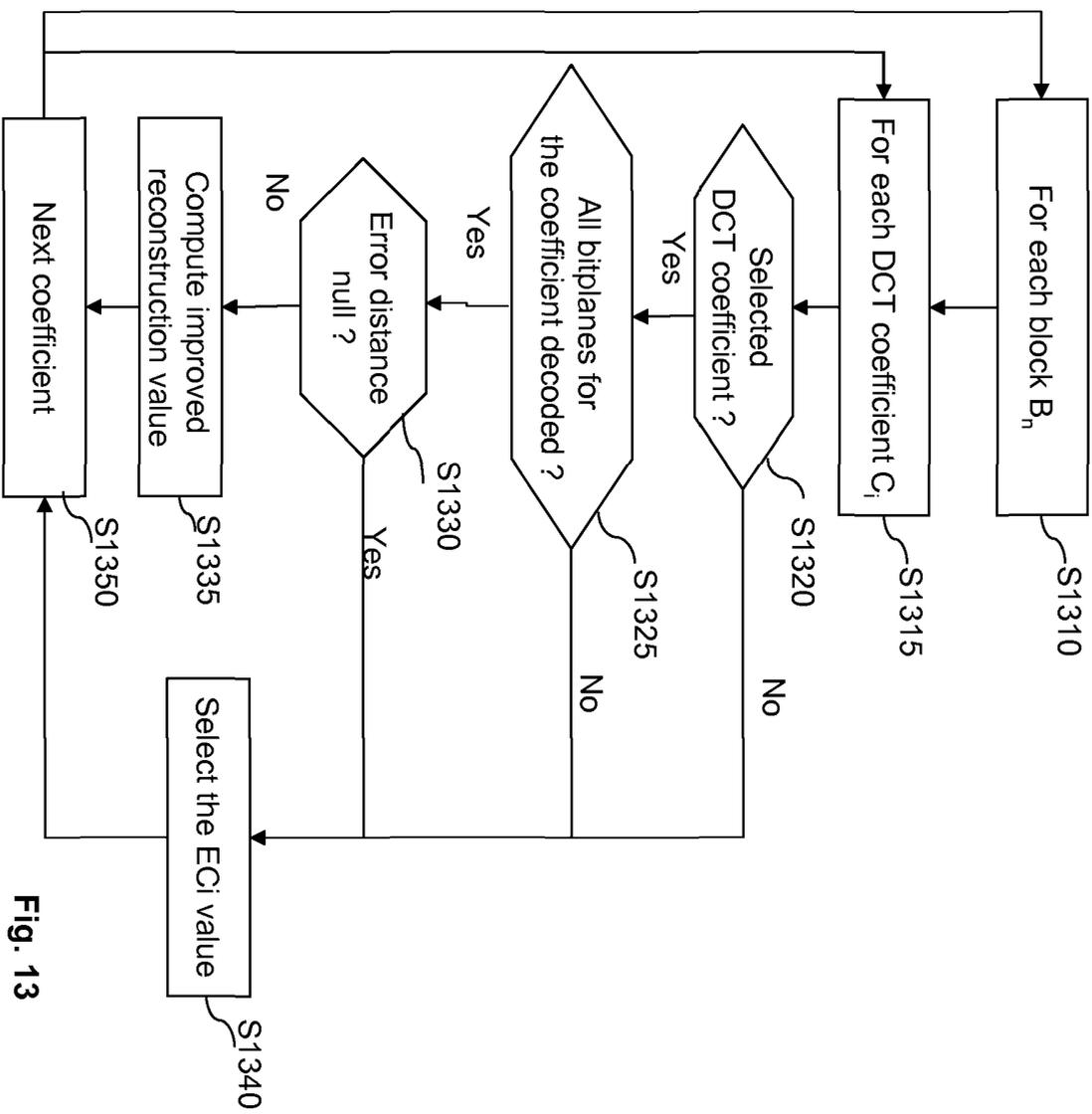


Fig. 13

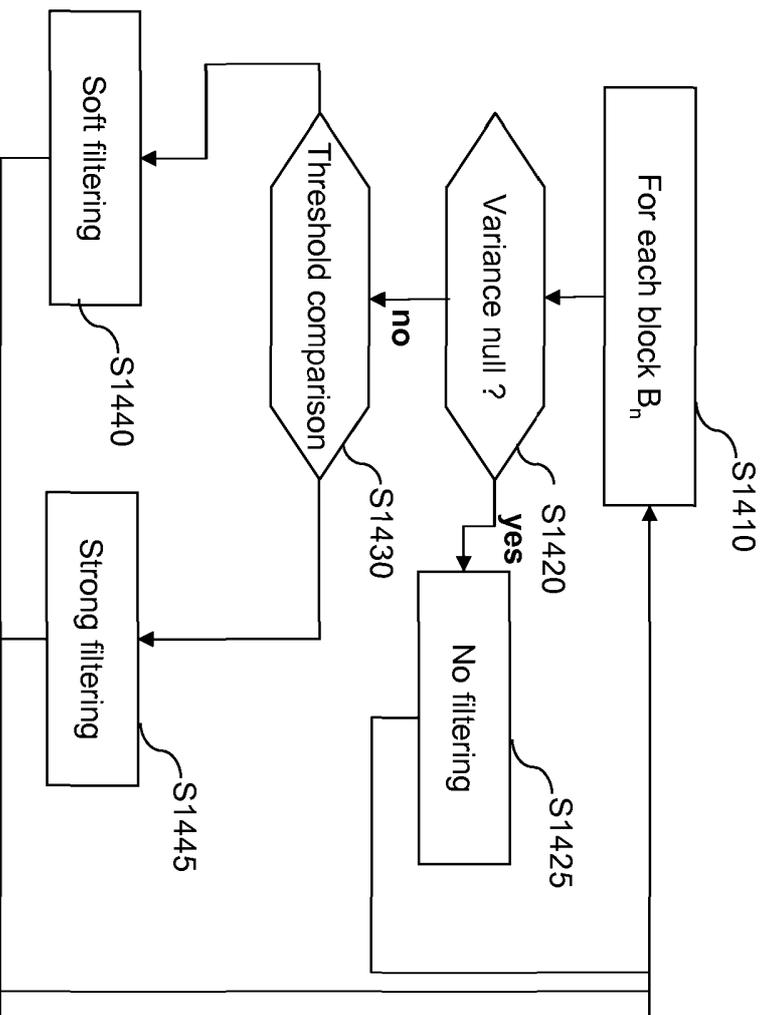


Fig. 14

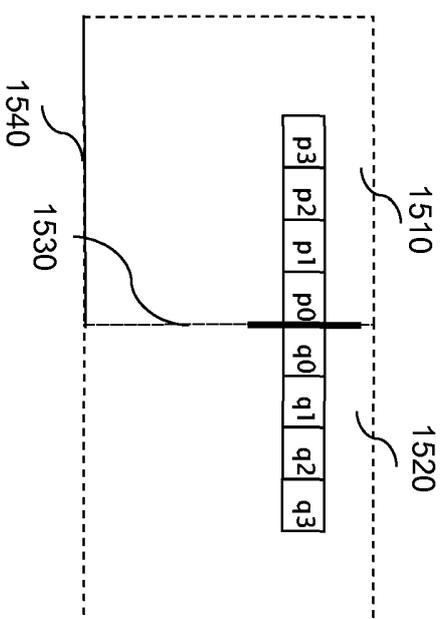


Fig. 15

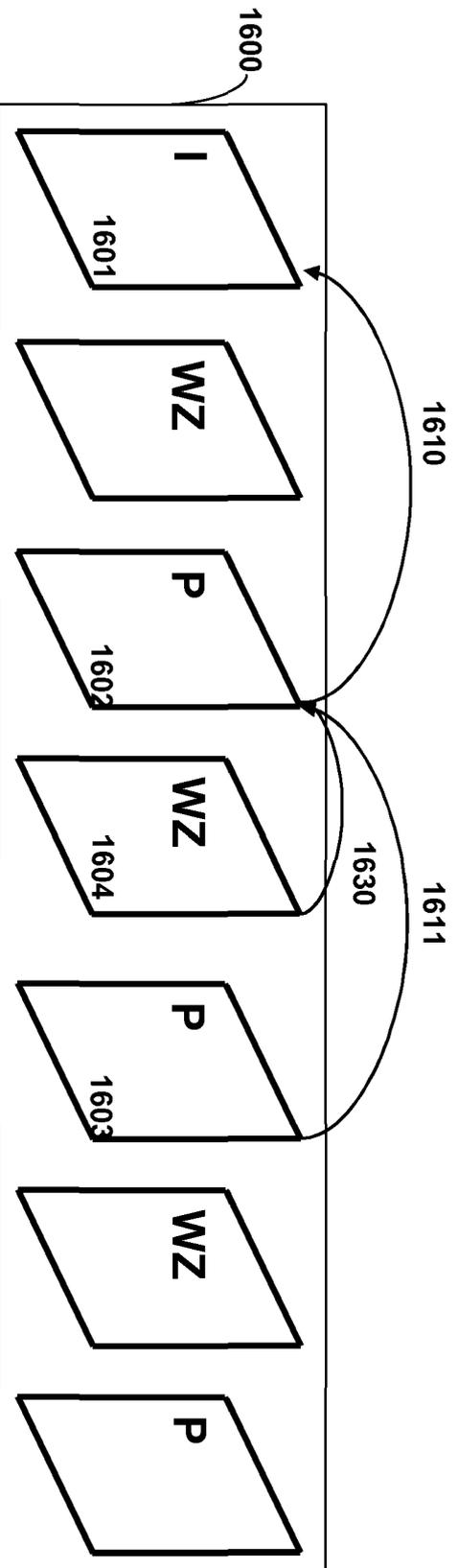


Fig. 16

Improved reconstruction of at least one missing area of a sequence of digital images

5

Field of the invention

The invention concerns a method and device for decoding a sequence of digital images containing at least one missing area.

10

The invention belongs to the domain of video processing in general and more particularly to the domain of decoding with reconstruction for missing parts, in particular using error concealment after the loss or corruption of part of the video data, for example by transmission through an unreliable channel.

15

Description of the prior-art

Most video compression formats, for example H.263, H.264, MPEG-1, MPEG-2, MPEG-4, SVC, called hereafter MPEG-type formats, use block-based discrete cosine transform (DCT) and motion compensation to remove spatial and temporal redundancies.

20

They can be referred to as predictive video formats. Each frame or image of the video sequence is divided into slices which are encoded and can be decoded independently. A slice is typically a rectangular portion of the image, or more generally, a portion of an image. Further, each slice is divided into macroblocks (MBs), and each macroblock is further divided into blocks, for example blocks of 8x8 pixels. The encoded frames are of two types: predicted frames (either predicted from one reference frame called P-frames or predicted from two reference frames called B-frames) and non predicted frames (called Intra frames or I-frames).

25

For a predicted P-frame, the following steps are applied at the encoder:

30

-motion estimation applied to each block of the considered predicted frame with respect to a reference frame, resulting in a motion vector per block pointing to a reference block of the reference frame. The set of motion vectors obtained by motion estimation form a so-called motion field;

35

-prediction of the considered frame from the reference frame, where for each block, the difference signal between the block and its reference block pointed to by the motion vector is calculated. The difference signal is called residual signal or residual data.

A DCT is then applied to each block of residual data, and then, quantization is applied to the transformed residual data, and

-entropic encoding of the motion vectors and of the quantized transformed residual data.

5 In the case of B-frames, two reference frames and two motion vectors are similarly used for prediction.

 For an Intra encoded frame, the image is divided into blocks of pixels, a DCT is applied on each block, followed by quantization and the quantized DCT coefficients are encoded using an entropic encoder.

10 In practical applications, the encoded bitstream is either stored or transmitted through a communication channel.

 At the decoder side, for the classical MPEG-type formats, the decoding achieves image reconstruction by applying the inverse operations with respect to the encoding side. For all frames, entropic decoding and inverse quantization are applied.

15 For Intra frames, the inverse quantization is followed by inverse block DCT, and the result is the reconstructed image signal.

 For predicted frames, both the residual data and the motion vectors need to be decoded first. The residual data and the motion vectors may be encoded in separate packets in the case of data partitioning. For the residual data, after inverse quantization,
20 an inverse DCT is applied. Finally, for each predicted block in the P-frame, the signal resulting from the inverse DCT is added to the reconstructed signal of the block of the reference frame pointed out by the corresponding motion vector to obtain the final reconstructed image block.

 A video bitstream encoded with such a predictive format is highly sensitive to
25 transmission errors, since an error will not only result in an incorrectly decoded image but will also propagate to the following images if the affected image is used as a reference image. In case of unrecoverable error, it is known, in video processing, to apply error concealment methods, in order to partially recover the lost or corrupted data, referred to as the missing area, from the compressed data available at the decoder.

30 It is also known in the prior art to use a so-called Wyner-Ziv scheme to reconstruct missing parts of a video encoded according to a predictive format. A Wyner-Ziv scheme can be applied in various video processing scenarios, such as distributed coding or error resilience.

 In a general view, the principle of a Wyner-Ziv scheme is to send from the
35 encoder to the decoder a small amount of data, known as Wyner-Ziv additional data, which can be used at the decoder to improve a part of the video sequence that can be

reconstructed or predicted from received video data. Typically, the additional data is composed of error correction symbols, such as FEC ("forward error correction") symbols, calculated for some subset representative of one or several image(s) of the video sequence.

5 One application of such a Wyner-Ziv scheme is error resilience, in case of transmission errors.

 The Wyner-Ziv scheme finds a second application in distributed video coding (DVC) : some frames of the video, which would have been encoded by prediction, are not generated at the encoder, but some additional Wyner-Ziv data representative of such
10 frames is transmitted. At the decoder, a missing frame is generated by prediction, and can be corrected using the Wyner-Ziv additional data.

 The article "Correlation Noise Modelling for Efficient Pixel Domain and Transform Domain Wyner-Ziv Coding" by Catarina Brites and Fernando Pereira, published in IEEE Trans. On Circuits and Systems for Video Technology, vol. 18, n° 9,
15 2008, describes a method for improving the quality of reconstruction of the predicted images in a Wyner-Ziv distributed video coding system. This article proposes to estimate the correlation between the predicted image and the original image, which is not known at the decoder, and to use this estimated correlation to improve the probabilities of a FEC decoder initialization. The correlation is estimated using the difference between two
20 predicted images, a backward prediction and a forward prediction.

 This method necessitates many calculations since two predicted images are computed and requires to have future images in order to compute the backward prediction, thus it is not adapted to a low delay context.

 The article "Optimal reconstruction in Wyner-Ziv Video Coding with Multiple
25 Side Information", by Denis Kubasov, Jayanth Nayak and Christine Guillemot, published in the 9th International Workshop on Multimedia Signal Processing 2007, describes a method for improving the predicted image values in a Wyner-Ziv scheme, based on a parameter evaluating the error between the predicted image and the original image, or noise correlation. It is supposed in this article that such a parameter is known for an entire
30 sequence of images, so no method for computing it is provided.

 This prior art improves on average the quality of the image reconstruction at the decoder, but does not take into account the spatially changing characteristics of a digital image signal.

SUMMARY OF THE INVENTION

5 It is desirable to remedy to some drawbacks of the prior art and to improve the quality of reconstruction of images of a video sequence in a Wyner-Ziv scheme, while reducing the computational complexity and adapting to the local characteristics of the images.

To that end, one aspect of the invention relates, according to a first aspect, to a method for decoding a digital signal comprising at least one digital image encoded by an encoder, said digital image being represented by a plurality of samples, the method
10 comprising, when a part of one said encoded digital image to decode is missing:

- applying a reconstruction to said encoded digital image having the missing part to form a reconstructed image, said reconstruction comprising setting a missing sample, being one of said samples in said missing part, to a predicted sample value,
15

-obtaining, for each of a plurality of predicted sample values, a value of an estimated error between said predicted sample value and the corresponding missing value, using additional data, derived by the encoder from at least part of the encoded digital image and usable during decoding to correct the encoded digital image,
20

- for at least one missing sample having a predicted sample value, computing a value representative of a confidence evaluation of the estimated error using said estimated error values obtained for a selected plurality of predicted sample values, and
25

-using said value representative of a confidence evaluation the estimated error obtained to improve the reconstruction of said encoded digital image to decode.

The method as described above advantageously improves the quality of image reconstruction in a Wyner-Ziv type scheme, by using a confidence evaluation of the
25 estimated error between predicted sample values obtained by reconstruction and original sample values, which are missing and unknown at the decoder. Advantageously, the confidence evaluation is computed individually for missing samples, depending upon a plurality of other predicted sample values, that is to say sample values obtained by
30 reconstruction, which can be chosen for example in a neighbourhood of the missing sample. Therefore, the method allows improving adaptively the error evaluation, which is representative of the correlation between reconstructed values and missing values. The digital signal is advantageously a video signal.

Several embodiments of the step of using said value representative of a
35 confidence evaluation obtained to improve the visual reconstruction quality of said

encoded image to be decoded, such as modifying the predicted sample values, and applying a corrective filtering of the reconstructed image.

According to an embodiment, the step of computing a value representative of a confidence evaluation of the estimated error comprises, for each said missing sample:

5 -selecting a plurality of samples according to a predetermined criterion, said plurality including said missing sample,

 -computing, as value representative of a confidence evaluation of the estimated error, a statistic of the estimated error values obtained for the predicted sample values of said selected samples.

10 Advantageously, a confidence evaluation of the estimated error based on a plurality of samples is computed, so as to limit the impact of isolated errors.

 According to an embodiment, the predetermined criterion is spatial proximity, so that local spatial characteristics of the digital image are taken into account.

 According to another embodiment, the missing sample belongs to a block of
15 samples and then predetermined criterion is a similarity between blocks of samples. In this case, the selected samples are close to the missing samples in terms of the similarity of content.

 According to an embodiment, the statistic is a variance of the estimated error values obtained for the predicted sample values of said selected samples. This allows
20 characterizing the distribution of the estimated error values among the plurality of samples selected.

 According to an embodiment, the step of obtaining a value of an estimated error comprises:

 -Applying an error correction decoding to said reconstructed image using said
25 additional data to obtain corrected symbols,

 -obtaining, for at least one missing sample, a corrected symbol representative of the missing sample value of said missing sample, and

 -computing an error distance between said corrected symbol and the predicted sample value of said missing sample.

30 In a particular case, the corrected symbol is representative of one or several intervals to which said missing value may belong, an interval being defined by its lower and upper bound, and the step of computing an error distance comprises:

 -for each said interval, computing the difference between said predicted sample value and the interval bound closest in value to said predicted sample value, the
35 error distance being set as the minimum of said computed differences.

The error correction decoding using additional data provides an efficient way of obtaining one or several possible intervals to which the original unknown value may belong. The amount of additional data is quite low, whereas the error estimation is quite accurate.

5 According to an embodiment, the error correction decoding is applied by bitplanes and comprises an iterative decoding process which uses, for each bit of a bitplane, a probability of said bit to be incorrect, and the method further comprises, to improve the reconstruction of said encoded digital image to decode, setting a probability of a bit to be incorrect based upon a value representative of a confidence evaluation of the
10 estimated error computed for a missing sample belonging to a previous image used as a prediction reference for said encoded digital image to decode.

Consequently, the values representative of a confidence evaluation of the estimated error are advantageously used to improve the error correction decoding.

15 According to an embodiment, which may be used in combination with the improvement of the error correction decoding, the step of using said value representative of a confidence evaluation obtained to improve the reconstruction of said encoded digital image to decode comprises, for at least one missing sample, a step of modifying the predicted sample value of said missing sample based upon the estimated error computed for said predicted sample value and upon the corresponding value representative of a
20 confidence evaluation of the estimated error.

According to an embodiment, wherein an encoded image is divided into blocks, the step of using said value representative of a confidence evaluation of the estimated error obtained to improve the reconstruction of said encoded image to decode comprises:

25 -applying a spatial filtering on samples situated at the borders of said blocks of said reconstructed image, said filtering consisting of applying one filter of a set of predetermined filters, comprising a step of selecting, among the set of predetermined filters, a filter to apply on a given border between a first and a second block, based upon said value representative of a confidence evaluation of the estimated error computed for a
30 sample of said first block.

All these ways of improving the reconstruction of the encoded image can be used either separately or in combination.

The invention also concerns, according to a second aspect, a device for decoding a digital signal comprising at least one digital image encoded by an encoder, said digital image being represented by a plurality of samples, the device comprising,
35 when a part of one said encoded digital image to decode is missing:

- means for applying a reconstruction to said encoded digital image having the missing part to form a reconstructed image, said reconstruction comprising setting a missing sample, being one of said samples in said missing part, to a predicted sample value,

5 -means for obtaining, for each of a plurality of predicted sample values, a value of an estimated error between said predicted sample value and the corresponding missing value, using additional data, derived by the encoder from at least part of the encoded digital image and usable during decoding to correct the encoded digital image,

10 - means for computing, for at least one missing sample having a predicted sample value, a value representative of a confidence evaluation of the estimated error using said estimated error values obtained for a selected plurality of predicted sample values, and

15 -means for using said value representative of a confidence evaluation the estimated error obtained to improve the reconstruction of said encoded digital image to decode.

The device according to the invention comprises means for implementing all the steps of the method for decoding a digital signal according to the invention as briefly described above.

20 The invention also relates to a carrier medium, such as an information storage means, that can be read by a computer or a microprocessor, storing instructions of a computer program for the implementation of the method for decoding a digital signal as briefly described above.

25 The invention also relates to a computer program which, when executed by a computer or a processor in a device for decoding a digital signal, causes the device to carry out a method as briefly described above.

The particular characteristics and advantages of the decoding device, of the storage means and of the computer program being similar to those of the method of decoding a digital signal, they are not repeated here.

30

BRIEF DESCRIPTION OF THE DRAWINGS

Other features and advantages will appear in the following description, which is given solely by way of non-limiting example and made with reference to the accompanying drawings, in which:

5 -Figure 1 is a schematic example of a communication system in which the invention can be implemented;

-Figure 2 illustrates a block diagram of a client device adapted to incorporate the invention;

-Figure 3 illustrates a block diagram of a server and a client in a first embodiment of the invention;

10 -Figure 4 illustrates an embodiment of auxiliary data extraction and generation;

-Figure 5 illustrates a typical distribution of the values of a given DCT coefficient in a current image;

15 -Figure 6 represents graphically a parity check matrix H of a very simple linear code of size $(7,4)$;

-Figure 7 is a flowchart of an embodiment of an error correction decoding according to an embodiment of the invention;

-Figure 8 is a flowchart of an embodiment of the computation of an estimation of error evaluation;

20 -Figure 9 is a flowchart illustrating an embodiment of the computation of an error distance map,

-Figure 10 illustrates the computation of the error distance for a coefficient value,

25 -Figure 11 is a flowchart illustrating an embodiment of an estimation of error confidence evaluation,

-Figures 12a and 12b illustrate examples of block neighborhoods,

-Figure 13 is a flowchart of a first embodiment of an improvement of the reconstruction quality of the decoded image,

30 -Figure 14 is a flowchart of a second embodiment an improvement of the reconstruction quality of the decoded image,

-Figure 15 illustrates pixel values of two neighbouring blocks, and

-Figure 16 represents describes a distributed video coding system in which a second embodiment of the invention can be applied.

35

DETAILED DESCRIPTION OF THE EMBODIMENTS

Figure 1 represents a schematic example of a communication system 10 in which the invention can be implemented. The communication system 10 comprises a server device 101 which is adapted to transmit data packets of a data stream to a receiving device or client device 102, via a communication network 100.

The communication network 100 may for example be a wireless network (Wifi / 802.11a or b or g), an Ethernet network, or the Internet network or a mixed network composed of several different networks.

As the transmission over the communication network 100 is not reliable, some errors can occur during the transmission. In particular, data packets may be lost, in case of congestion or interferences.

The system 10 may be a broadcast system, in which server 101 sends data streams to a plurality of client devices 102 at the same time.

In an application scenario of the invention, the data stream sent between the server 101 and the client 102 is a video sequence, encoded according to a predictive encoding format using motion compensation such as H.264 or MPEG-2. These formats provide compressed video data according to distortion-rate criteria, so as to provide videos at bitrates compatible with the bandwidth actually available on the network 100.

The encoded data are divided and encapsulated into transmission packets which are transmitted to the client 102 by the network 100 using a communication protocol, for example RTP (Real-time Transport Protocol) over UDP (User Datagram Protocol).

The client device receives the transmission packets, extracts data from the received packets to form the encoded stream and then decodes the stream to obtain decoded data, which can be either displayed or provided to a client application.

In case of transmission errors over the unreliable network 100, the client device applies error concealment to improve the quality of the decoded data. The embodiments of the invention as described below can be advantageously implemented by a client device 102 to enhance the quality of the decoded video data with reconstruction of missing parts.

Figure 2 illustrates a block diagram of a device, in particular a client device 102, adapted to incorporate the invention.

Preferably, the device 102 comprises a central processing unit (CPU) 201 capable of executing instructions from program ROM 203 on powering up, and instructions relating to a software application from main memory RAM 202 after the powering up. The main memory 202 is for example of Random Access Memory (RAM)

type which functions as a working area of CPU 201, and the memory capacity thereof can be expanded by an optional RAM connected to an expansion port (not illustrated). Instructions relating to the software application may be loaded into the main memory 202 from a hard disk (HD) 206 or from the program ROM 203 for example. The software application or computer program, stored on non-transitory computer-readable carrier medium, when executed by the CPU 201, causes the steps of the flowcharts shown in figures 3, 4, 7, 8, 9, 11, 13 and 14 to be performed on the client device 102.

A network interface 204 allows the connection of the device to the communication network. The software application when executed by the CPU 201 is adapted to receive data streams through the network interface from other devices.

A user interface 205 displays information to, and/or receives inputs from, a user.

Figure 3 illustrates a block diagram of a server and a client in a first embodiment of the invention. This embodiment applies to the improvement of the reconstruction of missing data due to transmission losses using error concealment. The processing is represented by connected modules, each module being adapted to implement, for example in the form of programming instructions, a corresponding step of a method implementing an embodiment of the invention.

The server device 101 receives from an external source, such as a camcorder for example, a sequence of digital images 30 to be encoded by an encoder 305. Alternatively, the sequence of digital images may have been stored in a memory of the server device 101 before processing.

The encoder 305 applies predictive coding using motion compensation according to one of the formats MPEG-2, MPEG-4 part 2 or H.264 and outputs a video bitstream 31. The video bitstream 31 is composed of units that can be decoded independently, called NALU (for Network Abstract Layer Units) in H.264 or slices in MPEG-4 part 2. In the subsequent description, the units that are encoded and decoded independently are referred to as slices.

The video bitstream 31 is transmitted to a client device 102 via the communication network 100. In this example, the video bitstream is encapsulated into RTP transmission packets which are sent via UDP.

The encoder 305 applies predictive type compression. Classically, in video compression standards (MPEG-2, MPEG-4 part 2, H.264) the images of a video sequence can be compressed according to one of the following modes: Intra mode (I), inter prediction mode (P) and bi-directional prediction mode (B).

A digital image is represented by a set of samples, also called pixels, each sample having an associated sample value. The samples representative of the image may be either pixel values in the spatial domain, or transform coefficients in a transform domain.

5 A colour image is represented with several components, for example Y, U, V. In the following description, the processing of a single component, the Y component, is described. The same processing can be applied similarly to all image components.

 In the Intra mode, an image is divided into blocks of samples (typically, blocks of 8x8 pixels). A transform is applied on each block, for example a Discrete Cosine
10 Transform (DCT). Next, quantization (Q) is applied. The quantization is a lossy transformation, since the values obtained after de-quantization (or inverse quantization) may be different from the values before quantization. After quantization, a lossless coding such as entropy coding is applied to obtain a bistream corresponding to an Intra image.

 In prediction modes, an image is also divided into blocks of samples, but each
15 block is encoded with respect to a reference block of a reference image, which has been encoded previously. In the Inter prediction mode (P), only one reference block from one reference image is used. In bi-directional prediction mode (B), a block is encoded with respect to two reference blocks, belonging respectively to two reference images. The reference block or blocks used for encoding a given current block are selected in the
20 reference image or images by motion estimation.

 The reference blocks are extracted from encoded/decoded reference images, so as to have the same reference blocks at the encoder and at the decoder in case of lossless transmission.

 For this reason, the server device 101 further comprises a decoder module
25 310 which carries out decoding of the video bitstream 31 encoded by the encoder module 305 generating such an encoded/decoded reference image.

 In the inter prediction mode (P), the reference image can be the previous image or another image of the video sequence which has already been coded. The reference block, selected by motion estimation, is subtracted from the block to encode,
30 and then the difference signal (also known as residual) is transformed using for example a DCT, then quantized. Finally, an entropy coding is applied to the quantized transformed residuals of a group of predicted blocks which are encoded as a slice.

 Further, the motion vector corresponding to each predicted block is also encoded in the video bitstream. For example, information identifying the reference image
35 (e.g. its temporal index) and the coordinates of the motion vector are encoded. The coordinates of the motion vector may be non integer ($1/2$ or $1/4$ of pixel).

Both for Intra and Inter modes, if an error occurs during the transmission of a slice, the whole slice is rendered impossible to decode. The number of slices in a video is not imposed by the standards. A low number of slices gives a better compression but a lower error resilience. For example ten slices could be created in a HD (High Definition) image.

The server 101 also generates additional data 32, which is derived from auxiliary data representative of some of the digital images of the sequence of images to be encoded, according to a Wyner-Ziv type encoding scheme. Auxiliary data (AD) is generated from an encoded/decoded image by modules 315 and 320, and then the additional data 32 to be transmitted to the client device is calculated from the auxiliary data.

Firstly, module 315 applies a transform (such as a DCT transform, similarly to the encoder 305) after a division into blocks of an encoded/decoded image to obtain blocks of transform coefficients. For example, for a block of 8x8 pixels, a set of 64 transform coefficients is obtained.

Next, a module 320 is dedicated to the extraction of auxiliary data from the blocks of transform coefficients.

The auxiliary data extraction will be described in detail with respect to figures 4 and 5. The result of the auxiliary data extraction module 320 is a set of symbols, the number of symbols being equal to the number of blocks obtained for the image to be processed, each symbols being represented with a predetermined number of bits per block.

Next, an error correction code of the symbols representing the blocks is computed by module 325. In the preferred embodiment, the Low Density Parity Check (LDPC) code is used.

In alternative embodiments, other error correction codes known in the art, such as turbo codes, can be applied.

An LDPC code is a linear block code. An error correction code can be characterized by the values (n,k) , where n is the number of symbols of a code word and k is the number of symbols of the information word. Knowing n and k , it is possible to compute the number of parity symbols $m = n - k$ and the code rate $R = k/n$. Typically in an LDPC code, the sizes k and m are very large.

In this embodiment, the LDPC code is applied on a subset of transformed and quantized coefficients of the image. For example, if the video is in HD format (High Definition), an image is represented by 1080x1920 pixels. An image is divided into blocks of 8x8 pixels on which a DCT is applied, resulting in 32400 blocks. In this case, $k=32400$

and using a code rate R of 0.91, we obtain $m=3240$ parity symbols. The advantage of using a very large size LDPC code adapted to the size of the image (typically, adapted to the number of blocks of the image) is that the spatial locations of the blocks are taken into account. For example, each block of quantized coefficients has an associated code symbol. Typically, the errors are spatially localized since they correspond to slices containing lost image data. A large size LDPC code makes it possible to correct badly concealed areas using the correctly received and decoded image data, for example corresponding to slices correctly received surrounding a lost or corrupted slice.

In this embodiment of a Wyner-Ziv type encoder, only the parity symbols (also known as check symbols) are transmitted as additional data 32 to the client device 102. In an embodiment, the additional data 32 is transmitted in a separate RTP stream, different from the RTP stream transporting the video bitstream 31. In an alternative embodiment, it is possible to integrate the additional data 32 within the video bitstream 31, for example using a SEI extension in format H.264 (standing for Supplemental Enhancement Information, which is additional non standard metadata defined in H.264 format for carrying enhancement information that can be used by a compliant decoder). In this alternative embodiment, a single data stream is transmitted from the server 101 to the client 102, containing both the video bitstream 31 and the Wyner-Ziv additional data 32.

The client device 102 receives the video bitstream 31 and the additional data 32. In practice, the data is received in the form of transmission packets through a network interface 204 (shown in figure 2), and a de-packetizer module (not shown in figure 3) extracts the data corresponding to the video packets from the transmission packets and the video packets containing the slice data are concatenated. The video bitstream is then decoded by the decoder module 330.

Slices received without error are processed by the decoder module 330 based upon the format of the bitstream.

For the slices containing at least one error, which are also referred to as corrupted slices, error concealment is applied.

Module 335 of the client device 102 implements an error concealment method EC to reconstruct of the missing data..

An error concealment EC is a reconstruction method which conceals the losses by using pixels obtained from correctly received data. Any suitable error concealment method may be implemented by module 335.

In a preferred embodiment, the error concealment method applied is motion extrapolation, which generally gives good results for sequences of images encoded according to a predictive format, in particular in the case of continuous motion with no

acceleration. In motion extrapolation, the motion vectors between the two previous images (I_{t-1}, I_{t-2}) of the video sequence are inverted to project part of the previous image (I_{t-1}) on the missing or corrupted parts of the current image (I_t).

Alternatively, another possible error concealment method is motion interpolation, which is efficient in presence of acceleration or change of motion in the image sequence. In motion interpolation, the motion vectors of the blocks belonging to the missing or corrupted area are calculated from the motion vectors of the surrounding blocks of the current image which have been received without errors.

After applying the error concealment method, a reconstructed image 33, which comprises a concealed or reconstructed part, is obtained.

However, even with a good error concealment method, the reconstructed image 33 is different from the corresponding original image at the encoder. In a classical system, the first decoded and concealed image 33 may create a bad visual effect when displayed, and because of the predictive encoding, the low visual quality may be propagated to the following images.

The additional data received can be used for an error correction decoding, to correct some parts of the decoded and concealed image and consequently to enhance the visual quality of the reconstructed image and to avoid error propagation to the following images of the sequence of digital images.

The decoded and concealed image 33 is then processed to extract auxiliary data, similarly to the processing applied at the encoder.

First, module 340 applies a block based DCT transform to obtain blocks of transform coefficients (34), similarly to module 315 of the encoder.

Subsequently, module 345 extracts decoding auxiliary data 35 from the transform coefficients obtained previously, similarly to module 320 of the encoder.

Module 350 applies error correction decoding on the extracted auxiliary data 35 using the received additional data 32, as described in more detail hereafter with respect to figures 6 and 7. In the preferred embodiment the additional data contains parity symbols of an error correction code of the auxiliary data. The error correction decoding module 350 provides corrected auxiliary data 36.

The corrected auxiliary data 36 and the transform coefficients 34 corresponding to the error concealed image 33 are then used by module 355 to estimate an error between the reconstructed data and the original missing data. Further, module 355 computes error evaluation values as described in more detail with respect to figure 8.

The error evaluation values are stored in an error evaluation map 37 which is representative of an evaluation of the confidence associated with the error concealment on the missing parts of the image. In other words, the error evaluation map can also be seen as representative of the correlation between the reconstructed sample values, also called predicted sample values, obtained by error concealment and the original samples values, which would have been obtained by decoding for the missing area if the data corresponding to the missing area had not been lost. The error evaluation map 37, as well as the error evaluation values, are then further used by the merge module 360 and/or by the filtering module 370 to enhance the quality of the reconstructed image..

The corrected auxiliary data 36 and the error evaluation map 37 are used to improve the transform coefficients obtained by error concealment by the merge module 360, as explained in detail with respect to figure 13. The merge module 360 outputs enhanced transform coefficients. Next, an inverse transform module 365 applies an inverse block DCT in this embodiment. An improved reconstructed image 38 is obtained.

Optionally, the improved reconstructed image 38 is further enhanced by the filtering module 370, which also uses the error evaluation map 37 to locally optimize the quality of the reconstructed image.

In an alternative embodiment, the filtering using the error evaluation map may be applied directly on the first reconstructed image obtained by error concealment.

The final resulting corrected image 39 may be output to an application (for example displayed), and also stored to be subsequently used as a reference image by the decoder module 330.

Figure 4 illustrates an embodiment of the auxiliary data extraction and generation applied by modules 320 at the encoder and 345 at the decoder.

The transform coefficients of the blocks of the current image are provided as an input.

At step S410, a subset of transform coefficients are selected for each block. For example, only a given number of coefficients corresponding to the low frequencies are selected. Classically, the transform coefficients can be ordered as a one-dimensional list of coefficients according to a so-called zig-zag scan, ordering the coefficients in the order of increasing frequencies and associating an index or number to each coefficient. The first N coefficients of the list are selected at step S410. In a preferred embodiment, N=3, so that coefficients of index $i \in \{C_i, 0 \leq i < 3\}$ are selected.

It is advantageous to select a given number of coefficients, in particular the first coefficients corresponding to the lower frequencies, since the visual impact of low frequencies is more important than the visual impact of high frequencies. Auxiliary data

selected in this way is therefore more efficient for correcting the error concealment result at the decoder. Moreover, error concealment generally better predicts the low frequencies than the high frequencies. Thus the auxiliary data selected in this way is better suited within the Wyner-Ziv scheme. In other words, for the same correction quality, the auxiliary data can be better compressed.

Next, at step S420, the extremum values, i.e. the minimum and maximum values for each selected coefficient, are determined. For each coefficient C_i of a given index i , $0 \leq i < 3$, the minimum (\min_i) and maximum (\max_i) values are computed considering all blocks of the image, and then a uniform quantization is applied between those minimum and maximum values at step S430. The number of quantization intervals is determined depending on the number M of bits chosen for representing each coefficient.

In the simplest implementation, an integer value representing an index of the quantization interval, also called quantization number (referral number 520 in figure 5), is directly encoded to represent the quantized value. In this case, in order to obtain a representation on M bits, the range between the minimum and maximum values is divided into 2^M quantization intervals.

However, in the preferred embodiment, the quantization number is encoded using co-set representation computed at step S440, and illustrated in figure 5. In this embodiment, in order to obtain a representation on M bits, $2^{M+1}+1$ quantization intervals are created between the minimum value \min_i and the maximum value \max_i for each coefficient. In the example of figure 5, $M=2$ and there are 9 quantization intervals with 4 different co-set numbers 0 to 3.

Note that at the decoder side, the minimum and maximum values per coefficient are not computed, but retrieved from the additional data 32 received, so as to ensure that the same quantization is applied at the encoder and at the decoder side. Alternatively, another parameter representative of the quantization interval for each coefficient, such as the quantization step, is transmitted to the client device along with the additional data 32. Similarly, the number of quantization intervals or alternatively the number M of bits per coefficient is known at the decoder side, being either transmitted or pre-determined.

Figure 5 illustrates a typical distribution of the values of a given DCT coefficient C_i in a current image. The horizontal axis 54 represents the values of the given coefficient C_i , which in practice vary between \min_i and \max_i . The vertical axis 52 represents the number of coefficients of index i , C_i , among the blocks of the current image, taking a value given by the horizontal axis. As shown in figure 5, a typical

distribution of an AC coefficient (coefficients of index $i > 0$), for example the first frequency coefficient C_1 , is centered around the value zero, i.e. in most blocks of an image, the value of C_1 is close to 0.

An uniform quantization is illustrated in figure 5: the range between \min_i and \max_i is divided into nine equal ranges or quantization intervals, each of which is attributed a different quantization number from 0 to 8. Each quantization interval is defined by its bounds, for example the lower bound z_5 and the upper bound z_6 of the quantization interval having quantization number 5 in figure 5.

The quantization intervals can be grouped into co-sets, which are attributed a co-set number varying between 0 and 3 in this example. The same co-set number is given to two or more quantization intervals, which advantageously achieves a more compact representation of the additional data derived from the auxiliary data. Therefore, an equal improvement can be obtained at the decoder with more compact additional data.

The number of quantization intervals of each co-set may be different. In the preferred embodiment, as illustrated in figure 5, the same co-set number 0 is associated with the middle quantization interval (containing the coefficient value 0) and to the two extreme quantization intervals. The remaining quantization intervals are grouped by two, so that co-set numbers 1, 2, 3 each have two associated quantization intervals.

As shown in the example, quantization sub-sets 1 and 5 are assigned the same co-set number 1. This has the advantage of reducing the number of bits for coding a value of a transform coefficient. However, the information cannot be decoded as such, as at the decoder, there would be an ambiguity as to how to decode a received co-set number. Supplementary information is necessary at the decoding.

In the Wyner-Ziv type encoding/decoding scheme represented in figure 3, such supplementary information is provided by the transform coefficients obtained by decoding and error concealment. Therefore, the co-set representation of the quantization intervals is well adapted in a Wyner-Ziv type encoding scheme for the representation of the auxiliary data. Advantageously, co-set representation of the quantization intervals allows compaction of the representation of the auxiliary data.

As further shown in figure 5, the co-set numbers can be simply encoded using the conventional binary representation. However, as shown in the figure, two consecutive values of the conventional binary representation may differ by more than one bit. However, when an error concealment is applied, often the predicted coefficient values are close to the original coefficient values, so generally, the quantized values are likely to be also close. Therefore, it is more advantageous to represent the quantized values with a code which has a Hamming distance of 1, meaning that two consecutive values differ only

by one bit. Such a binary representation is illustrated in figure 5 by the co-set Gray code 550.

Back to figure 4, the quantized values obtained at step S430 are represented with co-sets, computed as explained above with respect to figure 5. In step S440 a co-set number 530 is assigned to each co-set.. Next, at step S450, a binary encoding is applied on the co-set numbers obtained at step S440. In the preferred embodiment, a Gray code representation 550 is advantageously chosen.

Alternatively, a different type of quantization could be applied at step S430, for example a quantization with a dead zone around zero using 2^M quantization intervals.

Further, in an alternative embodiment, steps S440 and S450 could be replaced by a single step of binary encoding of the unique quantization numbers 520 associated with each quantization interval.

In another embodiment the step S450 could consist in shuffling the bits of the value obtained for each coefficient. A different shuffle order is applied in each block. However the order is selected deterministically so the decoder can unshuffle the bits of each coefficient. This embodiment has the advantage to give the same error probability for each bitplane and thus it can give a higher probability of decoding all the bitplanes. A bitplane is composed of all bits of same rank taken from each symbol, therefore each bitplane contains a bit per block.

An example of the error correcting coding module 325 and of the error correcting decoding modules 350 will now be given with respect to **figures 6 and 7**.

As already explained above, in the preferred embodiment, the LDPC (Low Density Parity Check) code is used by the error correction module 325 at the encoder to output the parity symbols which can serve for the correction of errors in auxiliary data.

Alternatively, other error correction codes such as turbo codes could be used.

An LDPC code is a linear code, defined by a sparse Boolean matrix H, also called the parity matrix.

Figure 6 represents graphically a parity matrix H, also known as LDPC matrix, of a very simple linear code of size (7,4), illustrated for the ease of explanation.

For example the matrix equivalent to figure 6 is $H = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$,

which is a Boolean matrix.

Given a word $\underline{c} = [c_1, c_2, c_3, c_4, c_5, c_6, c_7]$ composed of 7 symbols, it can be checked that the word is a code word by verifying that:

$$H\underline{c}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{Eq1})$$

where \underline{c}^T is transposed from vector \underline{c} .

It should be noted that the arithmetical operations are carried out as follows. The addition is the binary XOR operation and the multiplication is the binary AND operation. Note that addition and subtraction are the same operation and the inverse of a symbol is the symbol itself.

The parity matrix H is equivalent to the graph represented figure 6: the symbols of word c 610 are at the left of the graph and the check nodes p 620 which must be null are at the right. The symbols checked by each check node are linked to the check node. The links 630 between the symbols c_i and the nodes p_i of figure 6 represent the lines of parity matrix H.

For example, when applying the parity check (Eq1) with the first line of matrix H, the following relationship is obtained:

$$p_1 = c_1 + c_2 + c_5 = 0$$

When H has the form $\left[-R^T \mid I_{n-k} \right]$, i.e. the last n-k columns of H are equal to the identity matrix of size n-k, I_{n-k} , it is easy to compute the generating matrix $G = \left[I_k \mid R \right]$, where G is composed of the identity matrix of size k, I_k , and the transpose of the inverse of the first part of H, denoted R , of k lines and n-k columns. Given an information word $\underline{u} = [u_1, u_2, u_3, u_4]$, then $\underline{u} \cdot G = \underline{c}$.

The code is systematic: the first k values of the code word \underline{c} (c_1, c_2, c_3, c_4 in this example) are equal to the information word u composed of k information symbols. The last (n-k) values of \underline{c} (c_5, c_6, c_7 in this example) are the parity symbols which are transmitted as additional data 32.

In a regular LDPC code, the sparse matrix R is selected with the constraints that on each column the number of bits equal to 1 is fixed to a small value, for example equal to 3, so that each symbol is used for 3 checks.

In an alternative embodiment, an irregular LDPC matrix such as a Tornado code can be used, as described in Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, Daniel A. Spielman, Volker Stemann, "Practical loss-resilient codes", *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, p.150-159, May 04-06, 1997, El Paso, Texas, United States.

In the current embodiment, the matrix R is created and made available to the modules 325 and 350 before the video transmission. The matrix R depends on the number of blocks per image since in this embodiment, an information symbol is obtained from the quantized coefficients selected for a block.

5 In an alternative embodiment, it is possible to transmit from the server 101 to the client 102, along with the parity symbols data, either the matrix R or some parameters allowing the error correction decoding module 350 to generate exactly the same matrix R as used by module 325.

10 The encoding module 325 thus applies a multiplication by R to the binary encoded quantized data provided by module 320. In this operation, since an information symbol is formed by the selected quantized coefficients of a DCT block, all operations (XOR and AND) are thus applied per block and consequently, the encoding is very fast.

15 **Figure 7** is a flowchart of an embodiment of an error correction decoding method implemented by the error correction decoding module 350. All the steps of the algorithm represented in figure 7 can be implemented in software and executed by the central processing unit 201 of the device 102.

20 The error correction decoding module 350 implements an error correction decoding based on the probabilities of error per bit. In an embodiment, the algorithm applied is the algorithm called 'belief propagation', described in the article "Good error-correcting codes based on very sparse matrices", by D.J. C. MacKay, published in *IEEE Transactions on Information Theory* 1999, vol. 45, pp. 399-431.

25 The algorithm of figure 7 receives as an input (step S700) the concealment auxiliary data, (35 in the example of embodiment of figure 3), as well as the additional data received 32, which contains the parity symbols computed at the error correction encoding 325.

30 Next, at step S710, a bitplane to be processed is selected. Each symbol is encoded on a given number of bits, each bit having an associated rank from the most significant bit (MSB) to the least significant bit (LBS). For example, a symbol S encoded on 4 bits can be written as $S = b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0 = \sum b_i 2^i$, b_3 being the most significant bit of S and b_0 being the least significant bit of S .

35 A bitplane is composed of all bits of same rank taken from each symbol, therefore each bitplane contains a bit per block. In this embodiment, each symbol is composed of the binary representations of several quantized DCT coefficients. For example, if a symbol S encoded on 4 bits comprises two DCT coefficients C_1 and C_2 , then the first two bits (b_3, b_2) correspond to C_1 and the next two bits (b_1, b_0) correspond to C_2 .

If a Gray code was used as a binary representation of the quantized transform coefficients or if the bits of each coefficient have been shuffled, then all the bitplanes are independent. In that case, the bitplanes can be decoded in any order, so any bitplane that has not yet been decoded can be selected at step S710. If a conventional binary encoding was used for representing the quantization numbers, then the order of the bitplanes has an importance, and if the most significant bitplanes contain errors, the corresponding quantization number cannot be decoded. Therefore, in case a conventional binary encoding is used in step S450, the bitplanes must be selected at step S710 starting from the most significant bitplane to the least significant bitplane.

After the selection of a bitplane to be processed has been achieved at step S710, then at step S720 the probabilities per bit are computed for each bit of the selected bitplane, expressed as a probability ratio which is called the likelihood: for each bit, the probability of its value being 1 divided by the probability of its value being 0. As the algorithm progresses, these probability ratios will be modified taking into account information obtained from other bits, in conjunction with the requirement that the parity checks be satisfied.

Let p be the probability of a bit b to be incorrect.

The likelihood (or probability ratio) associated with the bit b is the following:

$$\text{if } b=1, \text{ likelihood}(b) = \frac{(1-p)}{p} \quad (\text{Eq2})$$

$$\text{if } b=0, \text{ likelihood}(b) = \frac{p}{(1-p)} \quad (\text{Eq3})$$

In the corrupted area of the signal, typically a slice that cannot be decoded and on which an error concealment is applied, the value of p is set to a predetermined value, for example $p=0.1$. Typically, in a simple embodiment, the same value of p is associated with every bit of every symbol representing a transform coefficient of a block belonging to the corrupted image area.

In the image areas where no transmission error occurred and thus the decoded value is correct, p is set to a value very close to 0, typically $p=0.0001$, such that the likelihood for a value $b=0$ is 0 and the likelihood for a value $b=1$ is the maximum acceptable integer value for the CPU 201.

With a conventional binary encoding there is a dependency between the bit correctness from different bitplanes. Thus if the decoding of a bit is incorrect in step S760, the lower order bits of the same DCT coefficient will be set to a likelihood of 1 to indicate a high probability of error (typically, $p=0.5$).

In an alternative embodiment, the likelihood calculation is improved using information on the predicted error values. This alternative embodiment takes into account the predictive encoding of images and the error propagation according to the motion. Therefore, it is desirable to take into account the estimated errors in the previous reference image to compute the initial likelihood of the bits of a current image.

This alternative embodiment advantageously uses the error evaluation map 37 computed for the previously processed reference image in order to improve the error correction decoding. In this alternative embodiment:

- if a block is lost, a constant bit error probability, $p=0.1$ is used;
- if a block is encoded in Intra mode, p is set to a value close to 0, for example $p=0.0001$;
- if a block is predicted via a motion vector from a predictor block of a reference image, a value representative of the estimated error (σ_B) on the predictor block is used. The computation of such a value σ_B will be described in more detail with respect to figures 9 to 11.

In case the predictor block is not aligned with the grid of blocks, the value σ representative of the estimated error of the block B of the grid of blocks which shares the largest surface with the predictor block is selected.

Then, for each DCT coefficient, the bit error probability may be computed according to the formula:

$$p = \frac{1}{\alpha_B} + \frac{\varepsilon}{(1 - e^{\alpha_B \varepsilon})} \quad (\text{Eq4})$$

with $\alpha_B = \sqrt{\frac{1}{2\sigma_B^2}}$ and $\varepsilon = 0.1$, and e^x designates the exponential

mathematical function.

The 'belief propagation' algorithm is applied at step S730.

For each parity check, the algorithm computes a probability ratio for every bit that participates in that parity check. These ratios give the probability of that parity check being satisfied if the bit in question is 1 divided by the probability of the check being satisfied if the bit is 0, taking account of the probabilities of each of the other bits participating in this check, as derived from the probability ratios for these bits with respect to this check.

For every bit symbol, the algorithm computes a probability ratio for each parity check in which said bit symbol is involved, giving the probability for that bit to be 1 versus

0 based only on information derived from other parity checks, along with the data received for the bit.

The algorithm alternates between recalculating the probability ratios for each check and recalculating the probability ratios for each bit.

5 At the end of each iteration, an evaluated value for each bit is computed from the likelihood.

A first simple embodiment consists to compare the likelihood to 1. If the likelihood is lower than 1 the bit value is estimated to be 0 otherwise, the bit value is estimated to be 1.

10 A better solution will be to test if the likelihood is greater than a threshold, for example $T1=1.5$, then the evaluated bit value is 1; otherwise, if the likelihood is lower than a second threshold (for example $T2=0.66$), then the evaluated bit value is 0. If the likelihood is between $T1$ and $T2$, then the bit value cannot be evaluated.

15 If the value of each bit of the selected bitplane can be evaluated, then the check values 620 are computed from the evaluated bit values using the classical LDPC decoding.

The algorithm stops (test S740) if all checks are verified ($Hc^t=0$), in which case the correction has succeeded and the bitplane is fully corrected, or after a predefined number of iterations, in which case the correction has failed. Typically, if the number of bit errors in the selected bitplane is sufficiently low, the 'belief propagation' decoding algorithm corrects all bits and stops rapidly. On the contrary, if the number of bit errors is too high, the 'belief propagation' decoding algorithm does not converge. The experiments have shown that for example a number of 20 iterations is sufficient.

25 If the algorithm has failed (answer 'no' to test S740), the selected bitplane is marked as undecodable at step S750. This means that the number of bit errors in the bitplane is high, and that the selected bitplane of the auxiliary data has not been completely corrected.

30 If the algorithm has succeeded (answer 'yes' to test S740), then the selected bitplane is completely corrected (S760). It is thus possible to find out exactly which bits of the selected bitplane of the auxiliary data provided as an input are incorrect. The incorrect bits are corrected by inverting their value in the auxiliary data (i.e. replacing value 0 by value 1 and vice versa).

35 In the case where the conventional binary representation is used, if a bit belonging to a given DCT coefficient is incorrect and has been corrected, the lower order bits of the same DCT coefficient have a high probability to be also incorrect. This information can be used to compute the likelihood for the corresponding bits at step S720.

Both steps S750 and S760 are followed by step S770, at which it is checked whether the currently selected bitplane is the last bitplane of the symbols of the auxiliary data.

5 If it is not the last bitplane, then the next remaining bitplane is selected and the steps S720 to S770 are repeated.

We note that if the conventional binary representation is used, and the processed bitplane has been marked as undecodable, then it is useless to process the following bitplanes of the DCT coefficient containing this selected bitplane. In this case and if this was the last DCT coefficient or if the currently selected bitplane is indeed the last bitplane of the auxiliary data, the error correction decoding stops (step S780).

Figure 8 is a flowchart of an embodiment of the computation of an estimation of error evaluation by module 355 of figure 3. All the steps of the algorithm represented in figure 8 can be implemented in software and executed by the central processing unit 201 of the device 102.

15 The algorithm of figure 8 receives as an input (step S800) the DCT coefficients computed from the reconstructed image by concealment (34 in the example of embodiment of figure 3), as well as the corrected auxiliary data 36 output by module 350. The DCT coefficients are predicted values for the original DCT values which are not known at the decoder. The predicted DCT value may be equal to the original DCT value, 20 in particular in the case where they belong to block correctly received at the decoder, and which are predicted from reference blocks which are also correct. Otherwise, the predicted values are likely to be different from the original unknown DCT value.

In step S810, a following DCT coefficient among the set of DCT coefficients $\{C_0, C_1, \dots, C_{N-1}\}$ that form the auxiliary data is considered, starting with the first DCT 25 coefficient C_0 , which is typically the DC coefficient.

In the preferred embodiment, an error evaluation map is computed per transform coefficient. This is advantageous since the transform coefficients may have different statistical distributions. In particular the DC coefficient has generally a different statistical distribution from AC coefficients.

30 However, alternatively, in a simpler embodiment, all selected coefficients may be processed together.

Next, it is checked at step S820 whether all the bitplanes, of the selected DCT coefficient for all the blocks of the current image have been corrected by the error correction decoding 350. In case of negative answer, the following DCT coefficient is 35 selected (S810).

In case of positive answer, at step S830 an estimated error value or error distance is computed for each block of the current image, and stored in a so-called error distance map. In this embodiment, the corrected auxiliary data is composed of corrected symbols that represent quantized values of DCT coefficients, so that an estimated error is computed for each block between the DCT coefficient obtained by error concealment, i.e. the predicted sample value, and the interval obtained from the corrected auxiliary data, as explained hereafter with respect to figure 10. Only an estimated error can be computed, since the information available is that the original DCT coefficient value, which is unknown, belongs to the interval obtained from the corrected auxiliary data.

Then, based on the estimated errors for a plurality of blocks, step S840 computes an error evaluation value associated with each block, finally obtaining an error evaluation map for the missing area. This computation is further detailed in figures 11 and 12. The error evaluation value associated with each block is a value representative of a confidence evaluation of the estimated error with respect to a plurality of blocks selected according to a predetermined criterion as explained further;

Step S840 is then followed by step S810 already described. If all the coefficients of the subset of coefficients selected to form the auxiliary data have been processed (answer 'no' to test S810), the algorithm stops (S850).

Figure 9 details an embodiment of the computation of an error distance map of step S830 of figure 8. All the steps of the algorithm represented in figure 9 can be implemented in software and executed by the central processing unit 201 of the device 102.

For each block of the current image (S900), the corrected auxiliary data value representative of a co-set number is decoded in step S910 to obtain a corresponding quantization interval, defined by its lower and upper bounds.

Considering the embodiment of figure 4, if a Gray code was used in step S450, firstly the Gray code value is decoded to obtain a co-set number 530. This step can be efficiently implemented by using a table, stored in memory, containing the correspondences between Gray code values and corresponding co-set number values.

A co-set number corresponds to several quantization intervals. For example, referring to figure 5, co-set number 1 corresponds to intervals 1 and 5. The bounds of the corresponding quantization intervals are considered, in this example $[z_1, z_2]$ and $[z_5, z_6]$.

For each such possible quantization interval, an estimated error distance is computed in step S920.

A quantization interval i , $[z_i, z_{i+1}]$ is represented in **figure 10**. The original DCT value, which is unknown at the decoder, is inside the quantization interval i . The DCT

value obtained by error concealment, or predicted sample value, is value y . In this embodiment, the estimated error distance γ_i is computed as follows:

If y is inside the interval, $z_i \leq y < z_{i+1}$, the estimated error distance is null :
 $\gamma_i = 0$;

5 Otherwise, the estimated error distance is the difference between value y and the closest interval bound: if $y < z_i$, $\gamma_i = z_i - y$; if $y > z_{i+1}$, $\gamma_i = y - z_{i+1}$.

If several possible quantization intervals are given by the co-set number, the minimum estimated error distance is selected (S930). This value of estimated error distance is then stored for the block (S940), along with an indication of the corresponding
 10 quantization interval.

When all the blocks have been processed, the algorithm stops. An estimated error distance map, comprising an estimated error distance for each block of the image for the given coefficient C_i is then available. It is worth pointing out here that due to the use motion compensation in the encoder, some reconstruction errors may be propagated from
 15 a previous reference frame, and therefore other image blocks than the blocks belonging to a missing area are likely to have associated error distances which are non null.

Figure 11 details an embodiment of the computation of an estimation of error confidence evaluation, or error evaluation map, of step S840 of figure 8. All the steps of the algorithm represented in figure 11 can be implemented in software and executed by
 20 the central processing unit 201 of the device 102.

For each block B_n of the current image (S1100) a value representative of a statistic of a plurality of estimated error values is computed in step S1120. Such a value is representative of an evaluation of the confidence on the estimated error for block B_n and for the given coefficient C_i .

25 In the preferred embodiment, the value representative of a statistic is the variance of the estimated error distance values obtained for the block B_n and for a plurality of other blocks of the current image, chosen according to a criterion. However, other statistical measures, such as higher order moments, may be alternatively used.

In an embodiment, the plurality of other blocks is chosen according to spatial
 30 proximity, so the blocks are selected in a spatial neighbourhood N_s of the block B_n .

The error variance σ_{B_n} of block B_n is computed as follows:

$$\sigma_{B_n} = \sqrt{\frac{1}{K} \sum_{B \in N_s} \gamma_B^2 - \left(\frac{1}{K} \sum_{B \in N_s} \gamma_B \right)^2} \quad (\text{Eq5})$$

where K is the number of blocks in the neighbourhood N_s including B_n , γ_B is the estimated error distance for block B and for the given coefficient, and the summations are carried out for all blocks B of the selected neighbourhood N_s , including the block B_n . We note that in an alternative, all coefficients are processed together, so in the formula of (Eq5), the estimated error distances for all coefficients used in the auxiliary data are taken into account.

Figure 12a shows a very simple example of spatial neighbourhood N_0 , comprising the 8 surrounding blocks of block B_n .

In an alternative embodiment, the algorithm tests a number of possible spatial neighbourhoods (four in the example illustrated in figure 12b, N_1 to N_4) by computing the corresponding variances and finally selects the neighbourhood the variance of which is the largest, N_4 in figure 12b. This alternative embodiment advantageously takes into account the local structure of the image signal, such as edges.

In yet another alternative, not illustrated in figure 12, a different criterion, for example signal similarity, may be applied to select the blocks to be used in the calculation of the variance. Such similar blocks may be situated at any position in the image, and are characterized by a similarity distance, such as for example a SAD (sum of absolute differences) computed pixel to pixel or coefficient to coefficient, is lower than a predetermined threshold.

More generally, the selection of blocks to be used in the calculation of the variance is applied according to a predetermined criterion, such as a proximity criterion, which is either spatial proximity or signal proximity or similarity.

Back to figure 11, once the value representative of a statistic of the error estimation for block B_n is computed, this value is stored in memory.

Once all the blocks have been processed, the error evaluation map 37 is completed.

Figure 13 is a flowchart of a first embodiment of an improvement of the reconstruction quality of the decoded image, as implemented by the merge module 360 of figure 3. All the steps of the algorithm represented in figure 13 can be implemented in software and executed by the central processing unit 201 of the device 102.

The algorithm uses as input the DCT values 34 computed for the reconstructed error concealed image 33, the error distance estimation and the error evaluation values, as well as the quantization intervals obtained from the corrected auxiliary data 36, obtained in step S910.

For each block B_n (S1310), and for each DCT coefficient C_i of such a block B_n (S1315), first test S1320 checks whether C_i is one of the coefficients selected to form the auxiliary data.

In case of negative answer, the DCT value obtained by error concealment is kept as final coefficient value in step S1340.

In case of positive answer, test S1325 verifies whether all the bitplanes have been corrected for C_i . In case of negative answer, step S1340 is applied, so the DCT value obtained by error concealment is kept as final coefficient value. In case of positive answer, in step S1330 it is checked whether the estimated error distance corresponding to coefficient C_i of block B_n , computed in step S930, is null.

If the estimated error distance is null, then the value obtained by error concealment is compatible with the error correction, and is therefore kept (S1340).

If the estimated error distance is non null, an improved value for the coefficient C_i of block B_n is computed in step S1335.

In the preferred embodiment, the improved value $x_{B_n,i}$ is calculated from the value $y_{B_n,i}$ which is the predicted DCT value obtained by error concealment for coefficient C_i of block B_n , using the bounds of the quantization interval $[z_i, z_{i+1}]$ obtained from the corrected auxiliary data, as proposed in the article 'Optimal reconstruction in Wyner-Ziv Video Coding with Multiple Side Information', by Kubasov et al., published in the 9th *International Workshop for Multimedia Signal Processing 2007*.

Let Δ be the quantization interval size: $\Delta = z_{i+1} - z_i$. The improved value is computed as follows:

$$\text{-if } y_{B_n,i} < z_i, \quad x_{B_n,i} = z_i + \frac{1}{\alpha_{B_n}} + \frac{\Delta}{1 - e^{-\alpha_{B_n}\Delta}} \quad (\text{Eq6})$$

$$\text{-if } y_{B_n,i} > z_{i+1}, \quad x_{B_n,i} = z_{i+1} - \frac{1}{\alpha_{B_n}} - \frac{\Delta}{1 - e^{-\alpha_{B_n}\Delta}} \quad (\text{Eq7})$$

The notation e^t designates the exponential mathematical function.

The value α_{B_n} is related to the value σ_{B_n} representative of an evaluation of the error confidence on a subset of estimated error distances computed for block B_n .

In the preferred embodiment, $\alpha_{B_n} = \sqrt{\frac{1}{2\sigma_{B_n}^2}}$. Alternatively, $\alpha_{B_n} = \sqrt{\frac{2}{\sigma_{B_n}^2}}$ can also be used.

The efficiency of using the formulas of (Eq6) and (Eq7) has been proven by experimental tests. If the variance σ_{B_n} of the estimated error distance values is high, the corresponding value of α_{B_n} is low, and the improved reconstruction value is close to the middle of the quantization interval. When the variance σ_{B_n} of the estimated error distance values is low, the corresponding value of α_{B_n} is high, and the reconstruction value is close to a bound of the quantization interval.

The value of α_{B_n} for the block B_n , for a given coefficient C_i , is calculated for block B_n , taking into account a plurality of the values of estimated error predictions for blocks in a given neighbourhood, in particular a given spatial neighbourhood. For this reason, the coefficient value improvement is advantageously adapted to the characteristics of the image signal, and the level of reconstruction improvement is higher than provided in the cited prior art article 'Optimal reconstruction in Wyner-Ziv Video Coding with Multiple Side Information', by Kubasov et al., published in the 9th *International Workshop for Multimedia Signal Processing 2007*, where a same value α is used for all blocks in an image.

Once the improved reconstruction value for the current DCT coefficient C_i of the current block has been computed, it is stored as new DCT value in block B_n (S1335).

Then the next coefficient of the block is processed, and the process is repeated until all coefficients and all blocks have been processed.

The result is a set of improved DCT coefficient values, which are then provided to the module 365 of figure 3 to compute an inverse block DCT.

Figure 14 is a flowchart of a second embodiment of an improvement of the reconstruction quality of the decoded image, as implemented by the filtering module 370 of figure 3. All the steps of the algorithm represented in figure 14 can be implemented in software and executed by the central processing unit 201 of the device 102.

This filtering is applied in the pixel domain, after the inverse block DCT has been performed, and it aims to further enhance the image quality since some blocking effects might appear due to the Wyner-Ziv correction scheme. Alternatively, such a filtering could be used directly after the error concealment.

The filtering strength is set, in this embodiment, based upon the estimated errors and their confidence evaluation based on a plurality of blocks.

For each block of current image (S1410), it is first checked in step S1420 whether the variance of estimated error distances for the coefficient DC (C_0) is null. In this embodiment, only the DC coefficient is considered since the DC coefficient has a major

visual impact. If the error variance is null, no filtering is applied (step S1425), and the algorithm returns to S1410 to process the following block if any.

If the error variance is not null, step S1420 is followed by step S1430 which applies the comparison of a quantity based upon the estimated error variance to a threshold, so as to decide between two types of filtering detailed hereafter, called soft filtering and strong filtering.

Let σ_{B_n} be the error variance for the DC coefficient of current block, and, as previously: $\alpha_{B_n} = \sqrt{\frac{1}{2\sigma_{B_n}^2}}$ and $\Delta = z_{i+1} - z_i$.

If $\frac{1}{\alpha_{B_n}} + \frac{\Delta}{1 - e^{\alpha_{B_n}\Delta}} < \frac{\Delta}{4}$, then step S1430 is followed by the soft filtering applied in step S1440.

Otherwise, step S1430 is followed by the strong filtering applied in step S1445.

If the blocks of an image are processed in the lexicographical order, from the top left corner to the bottom right corner, then for each current block, a vertical border and the horizontal border remain to be processed. For example, for block 1510 schematically represented on figure 15, borders 1530 and 1540 are processed.

In step S1440, the soft filtering is for example first applied on the vertical border and then on the horizontal border.

By reference to figure 15, the vertical border 1530 between blocks 1510 and 1520 is processed as follows. Each group of 4 pixels p_0, p_1, p_2, p_3 is filtered using the values q_0, q_1, q_2, q_3 from the adjoining block 1520. The new values p_0s, p_1s, p_2s, p_3s are computed for example as follows:

$$\begin{aligned} p_0s &= (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) \gg 3; \\ p_1s &= (p_2 + p_1 + p_0 + q_0 + 2) \gg 2; \\ p_2s &= (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) \gg 3; \end{aligned}$$

The symbol \gg is used for the bit shifting, in other words $V \gg a$ is equivalent to a division of V by 2^a .

The same filtering is applied analogously on the horizontal border. Next, the following block, if any, is handled (step S1410).

The strong filtering of step S1445 is applied similarly to the soft filtering, respectively on the vertical and the horizontal border.

For example, the following formulas may be applied:

$$p0s = (p2 + 2*p1 + 2*p0 + 2*q0 + 2*q1 + q2 + 5) / 10;$$

$$p1s = (p2 + p1 + p0 + q0 + q1 + 3) / 5;$$

$$p2s = (2*p3 + 3*p2 + p1 + p0 + q0 + q1 + 5) / 9;$$

5

Step S1445 is also followed by step S1410 for the processing of the following block.

Other filtering coefficients may be selected and applied, the main characteristic being that the soft filtering modifies less the pixel values than the strong filtering and takes into account less pixels of the adjoining block.

10

Figure 16 describes a distributed video coding system to which a second embodiment of the invention can be applied. The objective of a distributed video coding method is to simplify the encoding by removing a part of the motion estimation. Indeed the motion estimation in the encoder uses a lot of computation.

15

In a standard video coding system, frames are encoded in Intra mode (I frames), or in predicted mode based on one reference frame (P frame) or two reference frames (B frame). In a distributed video encoder some frames may be encoded classically (frames 1601, 1602 and 1603) and some frames may be encoded in Wyner-Ziv mode (frame 1604).

20

In this example, frame 1604 is encoded as Wyner-Ziv additional data, which can be generated using a method similar to the method used in figure 3. The original image is transformed with a DCT transformation, then the auxiliary data is extracted and finally an LDPC is applied on all the quantized blocks. Only the Wyner-Ziv additional data is transmitted to the decoder.

25

With this method the encoder is simplified compared to a classical encoder because no motion vector needs to be computed for a Wyner-Ziv encoded frame (noted WZ frame on the figure). However some motion vectors are computed for the classical P frames (1602 and 1603).

30

At the decoder side, the classical I and P frames (1601, 1602, 1603) are decoded by a classical decoder.

The frames encoded with Wyner-Ziv or WZ frames can be considered as missing areas of the sequence of images. In this embodiment, the missing area is an entire frame.

35

For the WZ frame (1604) the decoder creates a first version of the frame using the received I and P frames, by applying a reconstruction method using available data, similarly to error concealment methods.

Many methods exist but we will describe two efficient methods based on predicted motion estimation.

The motion extrapolation (ME) method consists in using the motion vectors of the previous P image (1610) and then changing their orientation and scaling them to the distance between the WZ frame and the P frame. This new motion field (1630) is then used to predict the pixel values of the WZ image.

The motion interpolation (MI) method consists in using the previous and following frames (1602 and 1603). A motion field is computed between the two frames or the motion field 1611 determined by the encoder can also be used. Then the values of the pixels of the image 1604 are predicted by interpolation of the pixels of the two images using the computed motion.

These two reconstruction methods are similar the to error concealment method applied on missing areas. It is still desirable in this embodiment to improve the reconstruction of the missing part, here the WZ frame, and the algorithms described above apply similarly in this case.

An estimated error distance per block and per coefficient, and an associated error confidence evaluation can be computed and use to improve the reconstructed coefficient values, and further to improve the decoded image quality in the pixel domain by filtering, and also to propagate the error confidence evaluation in the error correcting decoding algorithm, using the methods explained above.

In the embodiments described above, a block DCT transform was used to obtain transform coefficients. However, any alternative block or subband transform, such as a wavelet transform, may be equally applied.

In the embodiments described above, the partially corrected data obtained is representative of quantization intervals corresponding to transform coefficients, such as DCT coefficients computed for a block of samples.

Other alternative embodiments may be envisaged, such as for example auxiliary data representative of pixels values instead of coefficient values.

More generally, any modification or improvement of the above-described embodiments, that a person skilled in the art may easily conceive should be considered as falling within the scope of the invention.

CLAIMS

1. Method for decoding a digital signal comprising at least one digital image encoded by an encoder, said digital image being represented by a plurality of samples, the method comprising, when a part of one said encoded digital image to decode is missing:

5 - applying a reconstruction to said encoded digital image having the missing part to form a reconstructed image, said reconstruction comprising setting a missing sample, being one of said samples in said missing part, to a predicted sample value,

10 -obtaining, for each of a plurality of predicted sample values, a value of an estimated error between said predicted sample value and the corresponding missing value, using additional data, derived by the encoder from at least part of the encoded digital image and usable during decoding to correct the encoded digital image,

15 - for at least one missing sample having a predicted sample value, computing a value representative of a confidence evaluation of the estimated error using said estimated error values obtained for a selected plurality of predicted sample values, and

-using said value representative of a confidence evaluation the estimated error obtained to improve the reconstruction of said encoded digital image to decode.

20 2. A method according to claim 1, wherein the step of computing a value representative of a confidence evaluation of the estimated error comprises, for each said missing sample:

-selecting a plurality of samples according to a predetermined criterion, said plurality including said missing sample,

25 -computing, as value representative of a confidence evaluation of the estimated error, a statistic of the estimated error values obtained for the predicted sample values of said selected samples.

30 3. A method according to claim 2, wherein said predetermined criterion is spatial proximity.

4. A method according to claim 2, wherein said missing sample belongs to a block of samples and wherein said predetermined criterion is a similarity between blocks of samples.

35

5. A method according to any of claims 2 to 4, wherein said statistic is a variance of the estimated error values obtained for the predicted sample values of said selected samples.

5 6. A method according to any of claims 1 to 5, wherein said step of obtaining a value of an estimated error comprises:

-applying an error correction decoding to said reconstructed image using said additional data to obtain corrected symbols,

10 -obtaining, for at least one missing sample, a corrected symbol representative of the missing sample value of said missing sample, and

-computing an error distance between said corrected symbol and the predicted sample value of said missing sample.

7. A method according to claim 6, wherein said corrected symbol is representative of one or several intervals to which said missing value may belong, an interval being defined by its lower and upper bound, and wherein the step of computing an error distance comprises:

15 -for each said interval, computing the difference between said predicted sample value and the interval bound closest in value to said predicted sample value, the error distance being set as the minimum of said computed differences.

8. A method according to any of claims 6 or 7, wherein said additional data comprises parity symbols obtained at the encoder by applying an error correction encoding to auxiliary data extracted from a digital image corresponding to said encoded digital image to decode.

9. A method according to claim 8, wherein said step of applying an error correction decoding comprises a step of extracting auxiliary data from said reconstructed image comprising the sub-steps of:

30 -dividing said reconstructed image into blocks of decoded pixels, and

-applying a transform on each block of decoded pixels to obtain a block of transform coefficients.

10. A method according to claim 9, wherein extracting of the auxiliary data from the reconstructed image further comprises:

35

-selecting a subset of transform coefficients of each block of transform coefficients, and

-quantizing each selected transform coefficient using a predetermined number of bits, a quantization number being associated with each quantized coefficient.

5

11. A method according to claim 10, further comprising associating a co-set number to a plurality of said quantization numbers.

12. A method according to claim 10 or 11, wherein said extracted auxiliary data is binary encoded using a Gray code representation.

10

13. A method according to any of claims 6 to 12, wherein said error correction decoding is applied by bitplanes and comprises an iterative decoding process which uses, for each bit of a bitplane, a probability of said bit to be incorrect,

15

the method further comprising, to improve the reconstruction of said encoded digital image to decode, setting a probability of a bit to be incorrect based upon a value representative of a confidence evaluation of the estimated error computed for a missing sample belonging to a previous image used as a prediction reference for said encoded digital image to decode.

20

14. A method according to any of claims 1 to 13, wherein the step of using said value representative of a confidence evaluation obtained to improve the reconstruction of said encoded digital image to decode comprises, for at least one missing sample, a step of modifying the predicted sample value of said missing sample based upon the estimated error computed for said predicted sample value and upon the corresponding value representative of a confidence evaluation of the estimated error.

25

15. A method according to any of claims 1 to 14, wherein an encoded image is divided into blocks, the step of using said value representative of a confidence evaluation of the estimated error obtained to improve the reconstruction of said encoded image to decode comprises:

30

-applying a spatial filtering on samples situated at the borders of said blocks of said reconstructed image, said filtering consisting of applying one filter of a set of predetermined filters, comprising a step of selecting, among the set of predetermined filters, a filter to apply on a given border between a first and a second block, based upon

35

said value representative of a confidence evaluation of the estimated error computed for a sample of said first block.

5 16. A method according to claim 15, wherein the set of predetermined filters comprises a soft filter and a strong filter.

10 17. Device for decoding a digital signal comprising at least one digital image encoded by an encoder, said digital image being represented by a plurality of samples, the device comprising, when a part of one said encoded digital image to decode is missing:

- means for applying a reconstruction to said encoded digital image having the missing part to form a reconstructed image, said reconstruction comprising setting a missing sample, being one of said samples in said missing part, to a predicted sample value,

15 -means for obtaining, for each of a plurality of predicted sample values, a value of an estimated error between said predicted sample value and the corresponding missing value, using additional data, derived by the encoder from at least part of the encoded digital image and usable during decoding to correct the encoded digital image,

20 - means for computing, for at least one missing sample having a predicted sample value, a value representative of a confidence evaluation of the estimated error using said estimated error values obtained for a selected plurality of predicted sample values, and

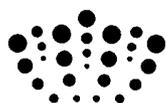
25 -means for using said value representative of a confidence evaluation the estimated error obtained to improve the reconstruction of said encoded digital image to decode.

30 18. A non-transitory computer program which, when executed by a computer or a processor in a device for decoding a digital signal, causes the device to carry out a method for decoding a digital signal as claimed in claims 1 to 16.

19. A computer-readable storage medium storing a program according to claim 18.

20. A method, device or computer program for decoding a digital signal as hereinbefore described with reference to the accompanying drawings.

35



Application No: GB1100224.3

Examiner: Mr Iwan Thomas

Claims searched: 1 and 17

Date of search: 7 May 2011

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

Category	Relevant to claims	Identity of document and passage or figure of particular relevance
A	-	Rane, S. et al., "Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting - a Wyner-Ziv Coding Approach", 2004, International Conference on Image Processing (IEEE), vol. 5, pages 3101-3104 See especially part 2.1
A	-	WO 2009/057956 A1 (SUNGKYUNKWAN UNIVERSITY FOUNDATION)

Categories:

X Document indicating lack of novelty or inventive step	A Document indicating technological background and/or state of the art.
Y Document indicating lack of inventive step if combined with one or more other documents of same category.	P Document published on or after the declared priority date but before the filing date of this invention.
& Member of the same patent family	E Patent document published on or after, but with priority date earlier than, the filing date of this application.

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC^X :

Worldwide search of patent documents classified in the following areas of the IPC

H04N

The following online and other databases have been used in the preparation of this search report

Online: WPI, EPODOC, Inspec

International Classification:

Subclass	Subgroup	Valid From
H04N	0007/68	01/01/2006