



(51) International Patent Classification:

H04N 19/645 (2014.01) H04N 19/94 (2014.01)

H04N 19/60 (2014.01) H04N 19/65 (2014.01)

(21) International Application Number:

PCT/GB2023/051588

(22) International Filing Date:

16 June 2023 (16.06.2023)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2208884.3 16 June 2022 (16.06.2022) GB

22179465.4 16 June 2022 (16.06.2022) EP

2305424.0 13 April 2023 (13.04.2023) GB

2305423.2 13 April 2023 (13.04.2023) GB

(71) Applicant: MBDA UK LIMITED [GB/GB]; Six Hills

Way, Stevenage Hertfordshire SG1 2DA (GB).

(72) Inventors: SHERRIFF, Andrew, John; MBDA UK Limited, PO Box 5, Filton, Bristol South Gloucestershire BS34 7QW (GB). VINEY, Matthew, Paul; MBDA, Golf Course Lane, Filton Bristol BS34 7QS (GB).

(74) Agent: BAE SYSTEMS PLC, GROUP IP DEPT; PO Box 87, Farnborough Aerospace Centre, Farnborough Hampshire GU14 6YU (GB).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(54) Title: METHOD FOR IMAGE ENCODING

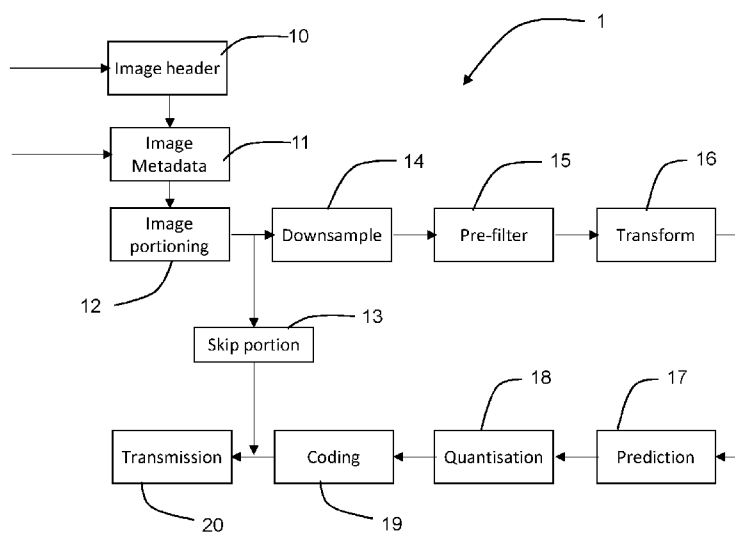


Fig 1A

(57) Abstract: A method for encoding data defining an image is disclosed. The method includes providing metadata associated with the image, encoding the metadata into binary code to form a metadata string, and repeating the metadata string a number of times. A method of decoding a bitstream to reconstruct an image is also disclosed. The method comprises identifying, in the bitstream, a metadata string containing bits relating to metadata associated with the image; determining the number of times the metadata string is repeated; and, for each bit in the metadata string, applying a voting procedure to determine the value of each said bit.

**(84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

**Published:**

— *with international search report (Art. 21(3))*

## METHOD FOR IMAGE ENCODING

### FIELD

The present invention relates to a method for encoding an image, for  
5 example to provide data suitable for wireless transmission. The invention  
further relates to a method of decoding such data.

### BACKGROUND

A number of methods for encoding image data are known. For example,  
10 the JPEG algorithm is widely used for encoding and decoding image data. In  
general the focus for such algorithms is the ability to retain high quality images  
whilst reducing the amount of data required to store the image. This reduction  
in the amount of data required to store an image results in more rapid  
transmission of images. Such compression algorithms are a key enabler for  
15 streaming of high quality video.

### SUMMARY

According to a first aspect of the present invention, there is provided a  
method for encoding data defining an image, the method including the step of  
20 providing metadata associated with the image, encoding the metadata into  
binary code to form a metadata string, and repeating the metadata string a  
number of times.

The metadata associated with the image may for example include a  
timestamp. The metadata associated with the image may for example include  
25 information relating to a subject of the image. The metadata may include any  
information relevant to interpretation of the image. Such information can be  
critical to later use of the image. Repeating the metadata string significantly  
enhances the resilience of the metadata to data loss errors that may occur  
during transmission of the image data.

30 The method may comprise the steps of:

- segmenting the image into image blocks, each image block in the  
portion having a uniform block size;

- applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;

- 5           - quantising the coefficients; and
- converting the quantised coefficients into binary code.

The metadata string may be repeated at least three times. The metadata string may be repeated at least five times, and preferably at least seven times. The more times the metadata string is repeated, the more resilient the metadata

10   information is to data loss.

According to a second aspect of the present invention, there is provided a method for encoding data defining an image, the method comprising the steps of:

- (a)     splitting the image into a number of image portions; and
- 15       (b)     for each of the image portions:
- segmenting the portion into image blocks, each image block in the portion having a uniform block size;
- applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is
- 20   represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;
- quantising the coefficients; and
- converting the quantised coefficients into binary code;
- each of the image portions being processed independently of the other
- 25   image portions.

Processing the image in separate, independent image portions enhances the robustness and resilience of the encoded data. Errors that may occur in transmission can at worst only propagate through one image portion, rather than through the whole image. Separating the image into portions can also

30   increase the processing speed for encoding the data, since the independent

portions can be encoded using a multi-threaded implementation. The separation of the image into image portions also enhances the flexibility of the encoding process.

In an example, for each portion, the uniform block size may be selected  
5 from a set of predetermined block sizes. Information signalling the block size to a decoder can be incorporated into a codeword defining an encoding mode. For any implementation of the encoding method, the number of encoding modes, as well as the actual block sizes used, can be configured as desired. Thus, for example, it may be appropriate to use larger block sizes in an  
10 implementation to be used for an image or image portion in which relatively uniform scenes are to be captured; or it may be appropriate to use smaller block sizes for images or image portions in which there are sharp edges.

The uniform block size for a first of the image portions may be different to the uniform block size for a second of the image portions. The encoding  
15 process can select an appropriate block size for each image portion.

The step of quantising the coefficients may be performed at a quantisation level that determines the resolution of the quantised data, and the quantisation level may be uniform for all the blocks in any one of the portions. Alternatively, the quantisation level for a first of the image portions may be  
20 different to the quantisation level for a second of the image portions. The quantisation level can therefore also be selected in dependence on the image or image portion to be encoded, capturing higher resolution as necessary or lowering resolution where it is more important to achieve high compression ratios for the encoded data.

25 The image may comprise a region of interest, in which case the method may further comprise the step of identifying a first of the image portions in which first image portion the region of interest is found; and a second of the image portions in which second image portion the region of interest is not found, and encoding the first image portion using a smaller block size and/or a finer  
30 quantisation level than those used for the second image portion. The method therefore enables the region of interest to be encoded appropriately with high resolution and detail, with other regions, for example, encoded with high compression ratios so as to maintain speed of transmission.

The method may further comprise the step of applying a pre-filter prior to applying the frequency-based transform, the pre-filter being applied to a group of pixels, and the group of pixels spanning a boundary between two image blocks. The pre-filter may for example mitigate artefacts in the reconstructed  
5 image arising from the application of the transform. In an example, the group of pixels is the same size as an image block.

The pre-filter may be determined at least in part by an optimisation process based on a set of selected images. For example, where the pre-filter is a matrix operation to be applied to the image data, one or more component  
10 parts may be optimised based on a set of selected images. The images can be selected to be of the same modality as those for which the pre-filter is to be used. In other words, if the pre-filter is to be used to encode infra-red images, the optimisation process can be based on a set of infra-red images. Likewise, if the pre-filter is to be used to encode images taken in the visible spectrum, the  
15 optimisation process can be based on a set of images taken in the visible spectrum. In this way the encoding method can be altered to suit the specific image modality it is to be used for, without the need to fully re-design the method.

In an example, the frequency based transform is a discrete cosine  
20 transform.

The method may further comprise the step of partitioning the blocks in each image portion into one or more sets of blocks, subsequent to the application of the frequency based transform. Partitioning the blocks reduces the potential for errors that occur in transmission to propagate throughout the  
25 whole image portion. For example, the blocks in each image portion may be partitioned into two or more sets of blocks.

The step of partitioning may be performed such that the blocks in any one of the sets do not share any boundaries. In one example there may be two sets of blocks, and the two sets may interlock. In such an example the two sets  
30 form interlocking 'checkerboard' patterns. In such a case, if an error or corruption during transmission results in loss of one set, it is easier for error concealment algorithms to recover at least some of the information from the lost set.

Each set of blocks comprises a plurality of slices of blocks, each slice consisting of a number of consecutive blocks in the set. The length of the slice may be uniform for all the image portions. Alternatively, slices in a first image portion may comprise a different number of blocks to slices in a second image portion.

Each slice may comprise a reference block, and the coefficients in subsequent blocks may be represented as a prediction based on the coefficients in the reference block. For example, the prediction may describe the subsequent coefficients as a difference from the reference value. Such prediction reduces the size of the data required to encode the image, but, if performed across a whole image or whole image portion, it will be seen that a single error in the reference block can propagate across the whole image, or image portion. By limiting the prediction to work across a single slice, errors are constrained to within that slice.

Each block may comprise one coefficient for a zero frequency basis function, and a plurality of coefficients for higher frequency basis functions, which plurality of coefficients for higher frequency basis functions are grouped into one or more sub-bands, each sub-band consisting of a number of coefficients.

The encoding can be performed using the coefficients for zero frequency basis functions only, and neglecting the coefficients for higher frequency basis functions. Useable information may still be obtained from the zero frequency coefficients only; and neglecting the higher frequencies results in a low amount of information being required to encode the image or image portion. For example, if the overall size of the encoded data is strictly limited, it may be possible for the encoding process to change to a mode in which only the zero frequency coefficients are encoded for some or all of the image portions.

The coefficients of a first sub-band in a subsequent block may be represented as a prediction based on the coefficients of said first sub-band in the reference block.

The coefficients for each of the one or more sub bands may be arranged in a predetermined order so as to form a vector, which vector has a gain and a

direction, and the direction of the vector may be quantised by constraining its component terms to be integers, and constraining the sum of those component terms to be equal to a predetermined value  $K$ . This provides an effective method for quantising the sub-band coefficients, which further enhances the compression ratios possible using the encoding method.

The encoding can be performed using the coefficients for zero frequency basis functions only, and the coefficients for higher frequency basis functions may be neglected. Useable information may still be obtained from the zero frequency coefficients only; and neglecting the higher frequencies results in a low amount of information being required to encode the image or image portion. For example, if the overall size of the encoded data is strictly limited, it may be possible for the encoding process to change to a mode in which only the zero frequency coefficients are encoded for some or all of the image portions.

The step of converting the quantised coefficients into binary code may comprise applying binary arithmetic coding using a probability model, and the probability model may be learnt based on a sample set of representative images. The probability model can therefore also be configured for use with specific image modalities.

The step of converting the quantised coefficients into binary code may comprise allocating bits associated with coefficients in each sub band in a slice amongst a set of bins in a predetermined order such that the bins each have substantially the same bin length; and the number of bins may be equal to the number of blocks in the slice. Fixing the length of the bins facilitates resynchronisation of the bit stream at the decoder in the event of data corruption during transmission. Limiting the application of the bit allocation scheme to working across a single slice enhances the resilience of the encoded data, since it limits the potential for an error to propagate. The length of the slice can be a configurable parameter for this reason, since shorter slices are more resilient to data corruption during transmission, but require greater processing power and bandwidth to encode.

The method may further comprise the step of interleaving the binary code. Interleaving may be performed in a separate dedicated transmission

apparatus, but, by incorporating the interleaving into the encoding process, increased resilience to burst errors is ensured.

According to a third aspect of the present invention, there is provided for encoding data defining an image, the method comprising the steps of:

5           (a)     segmenting the image into image blocks, each image block having a uniform block size;

             (b)     applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis  
10   functions having different spatial frequencies; such that each block of transformed image data comprises one coefficient for a zero frequency basis function, and a plurality of coefficients for higher frequency basis functions, which plurality of coefficients for higher frequency basis functions are grouped into one or more sub-bands, each sub-band consisting of a number of  
15   coefficients; and

             (c)     grouping the blocks of transformed image data into slices, each slice comprising a plurality of blocks of transformed image data; and

             (d)     converting the coefficients into binary code;

             wherein the step of converting the coefficients into binary code  
20   comprises applying an allocation method to allocate bits associated with coefficients in each sub-band in a slice amongst a number of bins such that the bins each have substantially the same predetermined bin length; the allocation method comprising the steps of: (i) assigning the bits associated with each sub-band in each slice to one of the bins; (ii) where the number of bits in a first bin is  
25   greater than the predetermined bin length, transferring bits to a second bin in which the number of bits is less than the predetermined bin length.

             Limiting the application of the bit allocation scheme to working across a single slice enhances the resilience of the encoded data, since it limits the potential for an error to propagate. The length of the slice can be a  
30   configurable parameter for this reason, since shorter slices are more resilient to data corruption during transmission, but require greater processing power and bandwidth to encode. Additionally, because the bit allocation scheme is

applied to sub-bands, rather than to entire blocks, the zero frequency coefficients are retained separately and can still be used in isolation to produce a decoded image (albeit of relatively lower quality) in the event that entire slices are corrupted during transmission.

5           The allocation method may be repeated iteratively. The allocation method may be terminated after a predetermined number of iterations have been completed. This ensures that the processing does not carry on indefinitely when only a small number of bits remain to allocate amongst otherwise substantially uniformly packed bins.

10           The number of bins may be equal to the number of blocks in the slice.

According to a fourth aspect of the present invention, there is provided a method for encoding data defining an image, the method comprising the steps of:

15           - segmenting the image into image blocks, each image block having a uniform block size;

            - applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;

20           - quantising the coefficients; and

            - converting the quantised coefficients into binary code

            wherein the step of converting the quantised coefficients into binary code comprises applying binary arithmetic coding using a probability model, and wherein the probability model is learnt based on a sample set of representative  
25           images.

            Where certain characteristics of an image to be encoded are generally known, training the probability model used for binary arithmetic coding on a sample set of images can lead to higher compression ratios than would otherwise be expected. Such characteristics might relate to the subject matter  
30           of the image; or may relate to the wavelength band at which the image is captured (the image modality). Thus, for example, the probability model for an

image taken from an airborne platform may differ from the probability model for an image taken at ground level in an urban environment. Similarly, the probability model for an infra-red image may differ from the probability model for an image obtained at visible wavelengths.

5           The probability model may be selected from a number of learnt probability models, each of the number of learnt probability models being learnt based on a sample set of representative images for a particular image modality. By storing such a number of learnt probability models, the encoding method can readily adapt to encode different image modalities. It may, for example, be  
10       possible to include a step in the encoding method to identify the image modality, and select the probability model to be used in dependence on the image modality. Alternatively the probability model can be selected by a user prior to beginning the encoding.

Where constraints are imposed on the values of the component  
15       coefficients for each vector representing sub-band coefficients, as has been described above, these constraints can also be used to inform the probability model. For example, the probability model may be a truncated normal distribution in the range between  $K$  and  $-K$  with variance  $\sigma$ , which variance is dependent on the number of components in the sub-band  $L$ , the predetermined  
20       value  $K$ , and the position  $i$  of the coefficient in the sub-band through the relationship:

$$\sigma = \alpha \left( \frac{K}{L-i+1} - \frac{K}{L} \right) + K\sigma_0 + \beta$$

in which relationship the parameters  $\alpha$ ,  $\beta$ , and  $\sigma_0$  for each sample set of representative imagery are calculated using a least-squares optimiser on the  
25       basis of the sample set of representative imagery. This model has been found to work well for medium wave infra-red imagery. The probability model may be the same for each sub-band. Alternatively, the probability model may be different for different sub-bands. The method may comprise learning the probability model for each sub-band separately.

30       As has been described above, the pre-filter may also be optimised for a specific image modality through the use of sample sets of reference images. This results in a flexible encoding method which, particularly in combination with

the learnt probability model, is particularly adaptable to different image types or modalities.

According to a fifth aspect of the present invention, there is provided a method for encoding data defining an image, the method comprising the steps of:

- segmenting the image into image blocks, each image block having a uniform block size;

- applying a pre-filter, the pre-filter being applied to a group of pixels, and the group of pixels spanning a boundary between two image blocks

- applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;

- quantising the coefficients; and

- converting the quantised coefficients into binary code

wherein the pre-filter is determined at least in part by an optimisation process based on a set of selected images.

The pre-filter may for example mitigate artefacts in the reconstructed image arising from the application of the frequency-based transform. Where certain characteristics of an image to be encoded are generally known, training the applying an optimisation process to determine, at least in part, the pre-filter, using a representative sample set of images can result in a more effective pre-filter for particular images. Such characteristics might relate to the subject matter of the image; or may relate to the wavelength band at which the image is captured (the image modality). Thus, for example, the pre-filter for an image taken from an airborne platform may differ from the pre-filter for an image taken at ground level in an urban environment. Similarly, the pre-filter for an infra-red image may differ from the pre-filter for an image obtained at visible wavelengths.

The images can be selected to be of the same modality as those for which the pre-filter is to be used. In other words, if the pre-filter is to be used to

encode infra-red images, the optimisation process can be based on a set of infra-red images. Likewise, if the pre-filter is to be used to encode images taken in the visible spectrum, the optimisation process can be based on a set of images taken in the visible spectrum.

- 5 Thus the optimisation, based on sample images, enables the pre-filter to be altered to suit images having those particular characteristics it is to be used for, without the need to fully re-design the method.

The group of pixels may be the same size as an image block.

- 10 Where the pre-filter is a matrix operation to be applied to the image data, one or more component parts may be optimised based on a set of selected images. The pre-filter may for example be defined by:

$$\mathbf{P} = \mathbf{W}_M \mathbf{D}_M \mathbf{W}_M$$

in which:

$$\mathbf{W}_M = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{J}_{M/2} \\ \mathbf{J}_{M/2} & -\mathbf{I}_{M/2} \end{bmatrix}$$

$$\mathbf{D}_M = \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{Z}_{M/2} \\ \mathbf{Z}_{M/2} & \mathbf{V} \end{bmatrix}$$

15

and in which  $\mathbf{I}_{M/2}$  and  $\mathbf{J}_{M/2}$  are  $M/2 \times M/2$  identity and reversal identity matrix respectively, and  $\mathbf{Z}_{M/2}$  is an  $M/2 \times M/2$  zero matrix, and where  $M$  is the width of the block; and wherein  $\mathbf{V}$  is a  $M/2 \times M/2$  matrix four by four matrix that is obtained by optimising with respect to coding gain, using suitable representative

- 20 The objective function may determine a metric related to the quality of the image. For example, the objective function may determine a level of noise in the transformed image data, such that, through an optimisation process, the level of noise can be minimised.

The objective function may be the mean square error:

$$\mathcal{L} = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W (I_{i,j} - \hat{I}_{i,j})^2$$

25

where  $I_{ij}$  are original image pixel values for a representative image,  $\hat{I}_{ij}$  are reconstructed pixel values, and  $H$  and  $W$  are, respectively, the height and width of the representative image in pixels; the reconstructed pixel values being those obtained after encoding an original image, exposing the encoded original image to a source of corruption to produce corrupted image data, and decoding the corrupted image data. Such an objective function takes into account factors arising from the encoding process and factors that may affect the image during transmission. As a result the use of an optimisation process based on such an objective function can enhance the robustness of the encoding process to specific transmission problems, particularly if such transmission problems are already known and can be modelled or repeated during the optimisation process.

According to a sixth aspect of the present invention, there is provided a method for encoding data defining an image, the method comprising the steps of:

- segmenting the image into image blocks, each image block having a uniform block size;
- applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;
- partitioning the blocks of transformed image data into one or more sets of blocks, and further partitioning each set of blocks into a plurality of slices of blocks, each slice consisting of a number of consecutive blocks in the set; wherein each slice comprises a reference block, and the coefficients in subsequent blocks are represented as a prediction based on the coefficients in the reference block;
- quantising the coefficients; and
- converting the quantised coefficients into binary code.

For example, the prediction may describe the subsequent coefficients as a difference from the reference value. Such prediction reduces the size of the

data required to encode the image, but, if performed across a whole image or whole image portion, it will be seen that a single error in the reference block can propagate across the whole image, or image portion. By limiting the prediction to work across a single slice, errors are constrained to within that slice. The  
5 resilience of the encoded image data is therefore enhanced at the cost of increasing the size of the data required to encode the image.

According to a seventh aspect of the present invention, there is provided a method for encoding data defining an image, the method comprising the steps of:

10 (a) segmenting the image into image blocks, each image block having a uniform block size;

(b) applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis  
15 functions having different spatial frequencies; such that each block of transformed image data comprises one coefficient for a zero frequency basis function, and a plurality of coefficients for higher frequency basis functions, which plurality of coefficients for higher frequency basis functions are grouped into one or more sub-bands, each sub-band consisting of a number of  
20 coefficients; and

(c) grouping the blocks of transformed image data into slices, each slice comprising a plurality of blocks of transformed image data; and

(d) concatenating the coefficients of a first sub-band of each block in a slice, converting the concatenated coefficients into binary code using binary  
25 arithmetic coding, and inserting an end-of-slice codeword at the end of the sub-band; and

(e) repeating step (d) for all the sub-bands in the slice, and then for all slices of the transformed image data.

The end-of-slice codeword supports the ability of a subsequent decoder  
30 to resynchronise, should an error arise as a result of loss or corruption during transmission. Linking the codeword position to the end-of-slice, rather than positioning a synchronisation code word arbitrarily within the bitstream, or at an

otherwise determined frequency, removes the need for further codewords to be added to indicate the end of a slice. Moreover by restricting the arithmetic coding to portions of sub-band data of only one slice in length, the potential for errors to propagate through the image is greatly reduced.

- 5           The method may further comprise the step of applying, subsequent to converting the coefficients to binary code using binary arithmetic coding, an allocation method to allocate bits associated with coefficients in each sub-band in a slice amongst a number of bins such that the bins each have substantially the same predetermined bin length; the allocation method comprising the steps
- 10 of: (i) assigning the bits associated with each sub-band in each slice to one of the bins; (ii) where the number of bits in a first bin is greater than the predetermined bin length, transferring bits to a second bit in which the number of bits is less than the predetermined bin length.

          Limiting the application of the bit allocation scheme to working across a

15 single slice enhances the resilience of the encoded data, since it limits the potential for an error to propagate. The length of the slice can be a configurable parameter for this reason, since shorter slices are more resilient to data corruption during transmission, but require greater processing power and bandwidth to encode. Additionally, because the bit allocation scheme is

20 applied to sub-bands, rather than to entire blocks, the zero frequency coefficients are retained separately and can still be used in isolation to produce a decoded image (albeit of relatively lower quality) in the event that entire slices are corrupted during transmission.

- 25           According to a eighth aspect of the present invention, there is provided a method for encoding data defining an image, the method comprising the steps of:

- segmenting the image into image blocks;
  - applying a frequency-based transform to each of the image blocks,
- 30 thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;

- converting the coefficients for each block into binary code, and concatenating the binary code for all of the blocks to form a bit stream;

- interleaving the bit stream to distribute the bit stream across a number of data packets; and

5           - storing the data packets.

Typically images are encoded for the purposes of transmission. The encoding can function to reduce the amount of data required to define the image. Distributing the bit stream across a number of data packets reduces the vulnerability of the encoded image to burst losses in transmission, since even  
10   loss of a number of consecutive data packets will not result in complete loss of a single part of the bit stream. The reconstructed bit stream is therefore more likely to be susceptible to error correction techniques. The interleaving may distribute the bit stream, for example, such that consecutive bits in the bit stream appear in different data packets.

15           By including the interleaving step in the encoding process, latency in any subsequent transmission is reduced. Typically a dedicated transmission apparatus may include an interleaving step prior to transmitting a data file. However, for this to be done, the transmission apparatus will need to first read the data file in order to be able to apply the interleaving step. When done as an  
20   integrated part of the encoding process, this additional pass through the data file is not needed. Overall, therefore, latency is reduced.

The step of interleaving may comprise writing the bit stream to an allocated memory store row-by-row, and reading data from the allocated memory store into the data packets column-by-column. Such an interleaving  
25   process can be referred to as a block interleaver. Each data column in the allocated memory store is longer than each data packet, such that each data packet contains null information. In this way the resilience of the encoded data is enhanced, since loss or corruption of the null information will not affect a reconstructed image.

30           The step of interleaving may comprise using a random interleaver. Such an interleaver distributes the bits from the bit stream randomly amongst the data packets.

Each data packet may be provided with header information comprising an image identifier, and an identifier to indicate the position of the data packet within the bit stream. For example, where the image is one of a number of frames in a video stream, the image identifier may indicate the frame number.

- 5           The method may further comprise the step of encrypting the data packets, the step of encrypting the data packets being performed prior to storing the data packets.

According to a further aspect of the present invention, there is provided a method of transmitting an image, comprising the steps of: receiving data  
10       defining an image from an image sensor; encoding the image according to the method described above; passing the stored interleaved bit stream to a transmitter; and transmitting the stored interleaved bit stream.

According to a further aspect of the present invention, there is provided a method of decoding a number of data packets to reconstruct an image, which  
15       image has been encoded according to the method described above. The method of decoding the number of data packets comprises inverting the steps described above. The method may comprise the steps of identifying, in the bitstream, a metadata string containing bits relating to metadata associated with the image; determining the number of times the metadata string is repeated;  
20       and, for each bit in the metadata string, applying a voting procedure to determine the value of each said bit.

According to a further aspect of the present invention, there is provided a platform comprising an image sensor, an image processor, and a transmitter, the image processor being configured to receive data defining images from the  
25       image sensor and encode the data according to the method described above, and to pass the data packets to the transmitter. The platform may be an aerospace platform, such as an unmanned air vehicle or a missile.

The transmitter may comprise a transmitter processor to receive the data packets, and a communications antenna to transmit the data packets.

30           The invention extends to a system comprising the platform, and a control station configured to receive the data packets and to decode the data packets according to the method described above. The control station may be a ground

station, or a vehicle, such as an aircraft, from which an operator can be in control of the platform.

The invention extends to a method of decoding a bit stream to reconstruct an image, which image has been encoded according to the method described above, the method of decoding comprising inverting the steps performed in encoding the image.

The invention further extends to a method of decoding a bit stream to reconstruct an image, which image has been encoded according to the method described above; wherein the method of decoding comprises inverting the steps performed in encoding the image; and the method further comprising the step of checking that the components terms sum to the predetermined value  $K$ . Implementing this check enhances the robustness of the decoding method.

If the component terms do not sum to the predetermined value  $K$ , an error may be identified. The error may, for example, be flagged to an error concealment algorithm.

If the component terms do not sum to the predetermined value  $K$ , the largest component term may be adjusted such that the component terms sum to the predetermined value  $K$ . Such an adjustment is likely to result in a value that is closer to the actual value, or at least to reduce the magnitude of the error.

The invention further extends to a method of decoding a bit stream to reconstruct an image, which image has been encoded according to the method described above; wherein the method of decoding comprises inverting the steps performed in encoding the image; and the method further comprises imposing a cap on the magnitude of predicted coefficients. The prediction process works to enhance the compression of the data because the variation in coefficients between adjacent blocks tends to be small. As a result, if a large prediction is read by the decoder, it is likely that an error has occurred. Implementing a cap on the predicted coefficients is therefore likely to reduce error, and enhances the robustness of the decoder.

The cap may be a fixed cap. Alternatively, the cap may be dependent on the magnitude of the reference coefficient. The cap may vary as a percentage of the reference coefficient, subject to a minimum value cap.

The invention extends to a method of encoding a series of image frames including at least a current frame and a preceding frame, each of the frames being encoded according to the method described above.

5 The invention further extends to a computer-readable medium having stored thereon data defining an image, which data has been encoded according to the method described above.

The invention further extends to a computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method described above.

10 The invention further extends to a processor configured to perform the method described above.

## BRIEF DESCRIPTION OF THE FIGURES

Embodiments of the invention will now be described by way of example only with reference to the figures, in which:

Figure 1a shows a schematic flow diagram illustrating a method of encoding data defining an image according to an example of the invention;

Figure 1b shows a schematic flow diagram illustrating a method of decoding a bit stream to reconstruct an image according to an example of the invention;

Figure 2 shows an image split into image portions in a step of a method according to an example of the invention;

Figure 3 is an illustration of the image of Figure 2 after downsampling in a step of a method according to an example of the invention;

25 Figure 4 shows the a portion of the image of Figure 3 segmented into blocks in a step of a method according to an example of the invention;

Figure 5 is a schematic illustration of the blocks of Figure 3 as transformed after application of a pre-filter and transform in a step of a method according to an example of the invention;

30 Figure 6 is an illustration of the partition of the transformed blocks of Figure 5 into two sets in a step in a method according to an example of the invention;

Figure 7 is an illustration of the grouping of the transformed blocks of one set of Figure 6 into slices in a step in a method according to an example of the invention;

Figure 8 is an illustration of how the coefficients in image blocks of different sizes can be scanned into a particular order in a step in a method according to an example of the invention;

Figure 9 is an illustration of the coefficients in one of the slices of Figure 7;

Figure 10 is an illustration to show how the zero frequency coefficients are predicted, quantised and converted to binary code in a step in a method according to an example of the invention;

Figure 11 is an illustration to show how the higher frequency coefficients are predicted, quantised and converted to binary code in a step in a method according to an example of the invention;

Figures 12 and 13 are flow diagrams illustrating further steps in the conversion of the coefficients to binary code in steps according to an example of the invention;

Figure 14 schematically illustrates a part of an interleaving process according to a step in a method according to an example of the invention;

Figure 15 is a graph comparing the performance of an example method according to the invention with the performance of known methods for encoding images;

Figure 16 shows an image encoded using a method according to an example of the present invention after reconstruction following transmission subject to varying bit error rates; and

Figure 17 is a schematic illustration of a platform and a ground station in communication to transmit an image encoded using a method according to an example of the present invention from the platform to the ground station.

## DETAILED DESCRIPTION

Embodiments of the present invention provide a method for encoding data defining an image to provide an image data file that offers increased robustness to data losses. Such data losses may occur as a result of wireless transmission of the image data file, and robustness to such data losses can

enable receipt of a useable image rather than total image loss. It will be understood that such robustness may result in a loss of eventual image quality when the image data file is decoded, although this is not necessary. By useable, it will be understood that the image data retains its integrity, such that the image can be reconstructed from the data and subsequently can be interpreted by a human operator, or by a computer performing a suitable image processing algorithm. Interpretation of the image may for example include detection or classification tasks, or any extraction of useful information from the reconstructed image.

## 1. OVERVIEW

Figure 1a is a schematic flow diagram 1 illustrating the steps performed in a method for encoding data defining an image. These steps will now be described at a general level, with further detail on their implementation provided in the following sections.

At step 10, an image header is provided. The image header contains the data defining the parameters used in the encoding process, and as such corruption in the image header can cause the complete loss of the image. The number of header bits is therefore kept small and of fixed length for each frame.

At step 11, metadata associated with the image is provided. The metadata includes information relevant to interpreting the image. For example, the metadata may include a timestamp indicating the time at which the image was captured; a frame number to indicate the relative position of the image in a sequence of images; information relating to how the image was captured, such as the waveband in which the image was captured, information identifying the sensor that captured the image and the parameters applied to the sensor during image capture; and/or information relating to preliminary image processing performed, such as information identifying a region of interest in the image (for example, a target or subject of the image identified by means of indicating the position and size of a box around the target or subject).

At step 12, the image is split into portions. An example of an image portion 210 is shown in Figure 2. Each image portion is processed

independently of the others in the subsequent encoding steps. This enables each image portion to be transmitted as soon as its encoding has completed, and so reduces latency. Since the image portions are processed independently, a useable image can still be obtained even if transmission losses result in complete failure for one image portion. Moreover, by splitting the image into portions, any errors arising from, for example, transmission losses, are constrained to be within one portion. This enhances robustness of the encoding/decoding process. In some cases, an image portion may be skipped, as illustrated at step 13. Each image portion for processing is further segmented into blocks.

At step 14, the image portion is downsampled. Downsampling reduces the information content in the image and can be done without significant loss of quality in the transmitted image. It should be noted that the amount of downsampling is dependent on the image being coded, and it will be possible to omit this step, particularly for relatively smaller-sized images. Pre-filters are optionally applied at step 15. The subsequent transform step can result in artefacts in the final image arising from the segmentation into blocks. As is explained in further detail below, the application of pre-filters can mitigate these artefacts. Of course, the pre-filter step can be omitted at the cost of retaining these artefacts.

At step 16, a transform is applied to each block. The transform is a frequency based transform, such as a discrete cosine transform. The purpose of the transform is to represent the image data as a linear combination of basis functions. The image data is thus transformed into a series of coefficients of different frequency basis functions. Frequency based transforms are typically used for image compression because in natural imagery information tends to be concentrated in low frequency components. Higher frequency components can therefore be stored at lower resolution, or often set to zero as a result of the subsequent quantisation step.

At step 17, prediction is performed. Typically, when ordered from low frequency to high frequency, the coefficients are highly correlated and this can be exploited by capturing the difference between one coefficient and the next,

rather than the actual coefficient itself. This is known as prediction, and can be used to compress the image data. Similarly, neighbouring blocks in images are also often highly correlated, and prediction can therefore be applied both within individual blocks and (particularly for zero-frequency coefficients) between blocks. In the event of transmission errors, the use of prediction can lead to significant problems, since loss of one coefficient results in loss of all coefficients predicted from that one coefficient. Prediction is therefore only applied to a limited extent to preserve resilience; or in some embodiments may be omitted.

At step 18, quantisation is performed. Quantisation further reduces the amount of data required to encode the image information by mapping the coefficients onto a limited number of pre-defined values. Various quantisation algorithms are known and can be used in the present method. Typically a quantisation level can be specified and varied, the quantisation level being, in broad terms, related to the resolution of the predefined values, and therefore to the amount of information compression that is achieved by the quantisation step. In one example quantisation scheme, coefficients for each basis function may simply be rounded. Other quantisation algorithms, described in further detail below, can also be used, and may retain a higher image output quality for a given amount of information compression, or have advantages in terms of robustness.

Encoding of the data into binary form is performed at step 19. Various methods are known for encoding data, such as variable length coding and fixed length coding. The coded data for the different blocks is multiplexed together. This results in a bit stream suitable for transmission at step 20. As is described in further detail below, a number of steps can be performed during coding to enhance resilience and robustness of the resulting bitstream. These can include application of error resilient entropy coding, and alternatively or additionally, interleaving the bit stream. The bitstreams for each of the image portions can be concatenated prior to interleaving. It should be noted that the interleaving can be integrated into the coding process, rather than being a step performed during transmission by a separate dedicated apparatus. Alternatively to immediate transmission, the bit stream may be stored in

memory, or another suitable storage medium, portable or otherwise, for decoding at a later point in time as may be convenient. It can be stored in a bespoke file format.

Decoding the bitstream, so as to obtain an image from the coded data, is  
5 achieved by reversing the steps outlined above. Additionally an error concealment algorithm may be applied as part of the decoding.

Figure 1b is a schematic flow diagram 5 illustrating the steps performed in a method for decoding data defining an image. The data is received and the image header is read at step 50. The image header contains information  
10 relating to the parameters needed by the decoder to decode the image. The image metadata is read at step 51.

At step 52, the binary code is translated to an appropriate form for subsequent processing, reversing the coding performed at step 19.

At step 53, any skipped image portions are replaced, for example (where  
15 the image is part of a sequence of images in video) with the corresponding image portion from a previous frame.

At step 54, any reconstruction necessary for quantised data is performed. If the quantisation is simple mapping of values to a constrained set, no reconstruction may be necessary. For more complex quantisation  
20 algorithms, however, such as the techniques described further below, some reconstruction may be necessary. As described further below, this step may assist in identifying any errors that have occurred during transmission or storage of the data.

At step 55, predicted values for coefficients are used to recover the  
25 actual values of the coefficients. This step simply reverses the prediction step used during encoding at step 17.

At step 56, the inverse of the frequency based transform is applied; and at step 57, a post filter is applied. The post filter inverts the pre-filter applied at step 15.

At step 58 error concealment can be applied. Error concealment may for example be based on values from neighbouring blocks where errors are detected; or may simply directly use values from neighbouring blocks.

At step 59, the data is upsampled as desired; and at step 60 the image  
5 portions are recombined to form the whole image.

## 2. PROCESSING COMPONENTS

An example of the invention provides a method of encoding and decoding (a codec) an image. The method of decoding an image follows the  
10 method of encoding an image, but in reverse. In the following, an exemplary method of encoding an image is described, with only the specific steps for decoding an image that differ from the reverse of the encoding method described.

### 2.1 Image Header

15 An image header is applied to the beginning of the coded data stream to determine the different configurable parameters that can be selected for coding the image. A small number of encoding modes are defined. Each mode specifies a different set of parameters determining how resilient the coded image is to data loss or corruption during transmission, and how much the  
20 image data will be compressed. The encoding mode may also specify, for example, whether or not the resulting coded image is to be of fixed or variable size; or whether individual image portions are to be of fixed or variable size. For example, eight different modes can be used. Fewer modes can be used, for example if image resolution and compression can be fixed; or more modes can  
25 be used if there is a greater variety of image resolution.

The image header includes an indication of which encoding mode is used. Where eight different modes are used, as in the present example, a binary codeword of only three bits are needed. This reduces the length, and therefore the potential for corruption, of the image header. This binary  
30 codeword can be repeated a fixed number of times, and a voting procedure applied to each bit in the binary codeword to ensure that the correct encoding

mode is used the vast majority of times. For example, the binary codeword may be repeated five or ten times. This enhances the robustness of the image code, since loss of the image header can result in complete loss of the image. Even with the use of repetition, the header may still be lost. However its likelihood is significantly reduced: for a bit error rate of one in one hundred bits (i.e.  $10^{-2}$ ), repeating the encoding binary codeword five times results in a likelihood of catastrophic image loss of roughly 1 in 1,000,000.

## 2.2 Image metadata

Metadata associated with the image can be provided from the image sensor itself, or from a processor associated with the image sensor. Such image metadata may include simple timestamps indicating the time at which an image was captured. However, as described above, the metadata may include any information associated with the image for the purposes of later interpretation of that image. Image metadata can be critical for later use of an image. It can be critical to know the time at which an image was captured, for example, if that image is to be used by a subsequent tracking algorithm to track a target's motion.

The image metadata is encoded as a bitstream separately from the image. It is repeated a number of times, for example five times. At decoding, a simple voting procedure can be used to ensure that each bit is correctly decoded. This can be the same as the voting procedure used for the header information. As with the header, the repetition of the metadata significantly reduces the risk of metadata loss.

## 2.3 Image Portioning

A received image is split into a number of image portions for subsequent processing. Each portion is a simple portion of the raw image data, comprising a strip of the image. Figure 2 shows an example image 200 split into a number of portions, such as portion 210. The size of the image portion is selected as to balance the competing requirements of latency, which is reduced as the image portion size becomes smaller, since the image portion can be transmitted as soon as its encoding is complete, and robustness, which can be reduced as the

image portion size is reduced and more portions are required to process the entire image. Whilst the use of image portions inherently increases robustness as a result of the constraining of errors to one image portion, rather than the whole image, use of too large a number of portions increases the likelihood of  
5 resynchronisation problems when errors occur (as each image portion is variable in terms of bandwidth).

Different encoding parameters can be specified for each image portion. For example, block size and quantisation level can be varied between portions. Changing the encoding parameters for particular image portions enables  
10 Region of Interest (ROI) coding. Portions which contain salient information can be encoded at a higher quality than those portions containing background information. To support this capability, it first needs to be understood which portions contain salient information. This can be achieved using existing image processing techniques to select the appropriate encoding parameters for each  
15 portion. Selected encoding parameters are provided to the decoder, for example by means of a header packet associated with each image portion. The image portion headers can also include the size, in terms of a number of bits, of each image portion. This results in a small increase in the amount of data required to transmit the information. In addition there is a risk of data corruption  
20 and consequent loss of useful image data, although, because of the use of image portions, any loss is isolated to the respective portion. Such risks can be mitigated by using repeated sending, and applying a voting procedure, as described above with reference to the image header.

It may be decided to entirely skip an image portion from subsequent  
25 processing in certain circumstances. This may be, for example, when very high compression ratios are desired, and it is possible to skip particular portions containing only limited salient information; or if the transmission channel is particularly noisy. When the processing of a particular portion is skipped, it can be replaced in the final coded data by the data from the previous image, if the  
30 image is part of a sequence of images forming a video feed; or it can simply be represented as blank. Before it is decided to skip an image portion, a metric is computed between frames to check the level of motion. If motion is negligible, then a skip portion can be selected by the encoder.

In subsequent processing, each of the image portions are processed independently. This supports resilience against data loss or corruption during transmission. In addition, the processing can be performed in a multi-threaded implementation, with each image portion being processed as an independent  
5 thread. For a multi-threaded implementation, the length of the encoded binary stream for each image portion can be included in the header information, so that each thread of the decoder knows which section of memory to read. In the case of a multithreaded implementation, once the portions are split, each portion is assigned to a thread. If there are more portions than threads, the  
10 portions may be queued for particular threads. The processing described in the following is done independently for each of the portions on different threads. The processing results in a bitstream for each of the image portions. These bitstreams can be concatenated prior to any interleaving step, which can enhance robustness as burst errors will be spread across a number of image  
15 portions, rather than affecting only one portion. It is also possible, subsequent to a portion encoding being completed, to immediately transmit the portion. In this case the bitstreams for each portion may be interleaved independently of the other portions prior to transmission. Such an implementation may increase processing speed by a factor up to the number of threads. Alternatively, the  
20 processing can be performed in a single thread. This can be beneficial for simplicity in some applications.

## 2.4 Down-sampling

Natural imagery exhibits a high degree of spatial redundancy. As a result in-loop filters and resampling techniques can be exploited to down-  
25 sample imagery/videos at the encoder, and then up-sample at the decoder with only small reductions in image quality. Down-sampling brings considerable benefits for data compression, since it results in smaller imagery, and therefore a smaller number of blocks need to be processed. Figure 3 shows an image 300 that is obtained by down-sampling on the sample image of Figure 2. Image  
30 300 is smaller than image 200. The amount of down-sampling can be configured in light of the type of images being processed. For example, an image of size 640 by 480 pixels may for example be down-sampled by a factor of 2 or 4. A greater down-sampling factor may be applied for higher resolution

images, or where a higher compression ratio of the image data for transmission is of greater importance. Any down-sampling factor can be applied as appropriate for the image being processed, and either integer or non-integer factors can be used.

- 5           In the present example, bicubic resampling is used. Bicubic resampling (see "Cubic convolution interpolation for digital image processing", IEEE Transactions on Acoustics, Speech, and Signal Processing **29** (6): 1153–1160) was found to provide a good balance between computational complexity and reconstruction quality.

## 10   2.5 Image Portion Segmentation

- Each image portion for processing is segmented into separate  $M \times M$  blocks of pixels. Segmenting reduces memory requirements, and limits the size of the visible artefacts that may arise due to compression and/or channel errors. An example of this segmentation process is shown in Figure 4, in which image
- 15   portion 400 is split into a number of blocks of uniform size with  $M$  equal to eight. It is possible to use different size blocks, or to adaptively select the block size. Smaller block sizes provide improved rate-distortion performance in areas with high change, such as at edges, whereas larger block sizes are preferred for flat textures and shallow gradients. Adaptively searching for the optimal
- 20   segmentation requires considerable computation time, and also limits robustness, since additional segmentation parameters must be passed to the decoder. In the present example these two requirements are balanced by selecting from a limited number of block sizes at the image portion level. Thus each image portion can have a different block size; and more specifically, in the
- 25   present example, the possible block sizes are  $M = 4$ ,  $M = 8$ , or  $M = 16$ . Each encoding mode uses a specific block size or combination of block sizes, and so block size information is encapsulated in the image header.

## 2.6 Pre/Post Filters

- Encoding algorithms that segment an input image into blocks can result
- 30   in artefacts in the image obtained on decoding the stored image. These artefacts occur especially at high compression ratios. It can be beneficial, both

perceptually and for algorithmic performance, if such artefacts are constrained to low spatial frequencies. This is because artefacts at low spatial frequencies do not typically impede either the functioning of algorithms for the detection of potential objects of interest, or the interpretation of the image by a human viewer. Both detection algorithms and human perception typically search for signatures at high spatial frequencies, such as shapes or edges.

Blocking artefacts occur particularly when the transform used is symmetric, as then block edges introduce strong discontinuities. When the inverse transform is applied during the decoding process, spatial errors introduced by the quantisation step cause misalignment in the block edges, resulting in visible artefacts in the decoded image. There are a number of ways of mitigating the problems associated with blocking artefacts. For example, lapped filters can be used before transformation. Lapped filters are filters that are applied across the block boundaries. A number of suitable filters exist. Alternatively, deblocking filters can be used during the decoding process. Deblocking filters, however, do not directly address the underlying issues that cause the artefacts. In the present example, therefore, a lapped filter is used.

In general terms, lapped filters function to alleviate the problem of blocking artefacts by purposely making the input image blocky, so as to reduce the symmetric discontinuity at block boundaries. When a suitable lapped filter is paired with a suitable transform, such as a direct cosine transform, the lapped filter compacts more energy into lower frequencies. A further benefit of the use of lapped filters is that the filter used can be designed specifically for the image modality (for example, infra-red images; synthetic aperture radar images, or images in the visible spectrum). Thus the image codec can be modified for a specific modality without a complete redesign of the codec being necessary.

In the present example a lapped filter  $\mathbf{P}$  is applied across  $M \times M$  groups of pixels throughout the image portion. Each group of pixels spans two neighbouring blocks. The structure of  $\mathbf{P}$  can be designed to yield linear-phase perfect reconstruction filter bank:

$$\mathbf{P} = \mathbf{W}_M \mathbf{D}_M \mathbf{W}_M$$

where:

$$\mathbf{W}_M = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{J}_{M/2} \\ \mathbf{J}_{M/2} & -\mathbf{I}_{M/2} \end{bmatrix}$$

$$\mathbf{D}_M = \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{Z}_{M/2} \\ \mathbf{Z}_{M/2} & \mathbf{V} \end{bmatrix}$$

$\mathbf{I}_{M/2}$  and  $\mathbf{J}_{M/2}$  are  $M/2 \times M/2$  identity and reversal identity matrix respectively, and  $\mathbf{Z}_{M/2}$  is an  $M/2 \times M/2$  zero matrix. Thus, for  $M = 8$ ,  $\mathbf{I}_{M/2}$ ,  $\mathbf{J}_{M/2}$ , and  $\mathbf{Z}_{M/2}$  are four by four matrices.  $\mathbf{V}$  is a four by four matrix that uniquely

specifies the filter. It can be refined for particular image types or image modalities, so that the filter can be tailored for the image type that the encoding is to be performed on. The matrix  $\mathbf{V}$  is obtained by optimising with respect to coding gain, using suitable representative imagery, and a suitable objective function. For example, the objective function may be the mean squared error:

$$\mathcal{L} = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W (I_{i,j} - \hat{I}_{i,j})^2$$

where  $I_{i,j}$  are the original, and  $\hat{I}_{i,j}$  the reconstructed image pixel values, and  $H$  and  $W$  are the height and width of the image in pixels respectively. The reconstructed image pixel values are those obtained after encoding, transmission and decoding. This exemplary objective function models the impact of channel distortions such as bit-errors end-to-end.

The optimisation can be performed by calculating the objective function for each block in a frame, and then calculating an average value for the frame.  $\mathbf{V}$  is determined as the four by four matrix which minimises the average value thus obtained. The optimisation can be extended to calculate an average of the objective function over a number of frames. It will be understood that such an optimisation may enhance resilience, since the objective function models channel distortions that impact the image during transmission.

By changing the modality of the representative imagery, the filter can be tailored to a particular image modality. Thus, to tailor the image for use with infrared images, the representative imagery can comprise infrared images; whilst for use with images taken in the visible spectrum, the representative  
 5 imagery can comprise images taken in the visible spectrum. In some cases, it may be possible to use images that are also representative of the subject of the images it is expected to apply the encoding method to. In other words, where it may be expected to use the encoding method to encode images taken in an urban environment, the representative imagery can be selected to be images of  
 10 an urban environment.

## 2.7 Transform

A two dimensional discrete cosine transform (DCT) is applied to the filtered blocks. In the present example a two dimensional DCT-II is used, and the coefficients  $B_{k_1, k_2}$  are accordingly computed as:

$$B_{k_1, k_2} = \sum_{n_1=0}^{M-1} \sum_{n_2=0}^{M-1} x_{n_1, n_2} \cos \left[ \frac{\pi}{M} \left( n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[ \frac{\pi}{M} \left( n_2 + \frac{1}{2} \right) k_2 \right]$$

where  $x_{n_1, n_2}$  is a coefficient at  $n_1, n_2$  in the block of size  $M$ , and  $k_1, k_2$  define the location of the coefficient  $B_{k_1, k_2}$  in the transformed block.

The basis functions are cosine functions with varying wavenumbers  $k_1, k_2$ . Application of the transform enables the energy of the block to be compacted  
 20 into only a few elements.

## 2.8 Implementation of Pre-filter and Transform

Calculation of the pre-filter and transform is the done in a single step using a lifting implementation, as described by Jie Liang et al. in 'Approximating the DCT with the lifting scheme: systematic design and applications',  
 25 Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat No.00CH37154), 2000, pp.192 – 196 vol. 1. The lifting implementation simplifies computation for both encoding, and for

decoding, during which the inverse is applied. It has benefits for numerical resolution, and may also enable fully lossless operation.

Whilst the lifting implementation reduces the number of computations dramatically, computing the DCT and pre-filter remains a computationally demanding process. For example, a single High Definition (HD) image frame (1920x1080 pixels) requires around 32,400 calls to the DCT and lapped filter function. For a 640x480 pixel image, around 4,800 calls are required. Approximate versions of the DCT can be used, and these may enable a reduction in the number of numeric operations. It is believed that computational complexity can be reduced by up to 50% using such approximations. Such methods can also be adapted specifically for FPGA exploitation.

## 2.8 Block ordering

The order in which the blocks are processed can be adapted in order to enhance the robustness of the codec. Enhanced robustness arises as a result of the order in which the prediction step is applied to the blocks, as is described in further detail below.

In the present example, the blocks are grouped into two interlocking sets. A first set comprises alternate blocks along each row of the image portion, and alternate blocks along each column of the image portion. A second set comprises the remainder of the blocks in the image portion. Thus the second set also comprises alternate blocks along each row of the image portion, and alternate blocks along each column of the image portion.

The two sets are schematically illustrated in Figure 6. As can be seen, the first set 610 and the second set 620 each form a checkerboard pattern. The first set and the second set interlock, and together include all the blocks in the image portion.

For subsequent processing the first and second sets are further partitioned into slices, each slice comprising a number of blocks. Figure 7 illustrates the partition into slices of a checkerboard pattern of blocks 700. By way of example the slices in Figure 7 each have four blocks. Slice 710 is highlighted. Conceptually, the slice is flattened such that the blocks are

adjacent to each other as illustrated. Larger slices result in better rate-distortion performance, whereas smaller slices better support resilience of the encoded image.

## 2.9 Prediction

5           For the prediction step, each block is further divided into a zero frequency, DC coefficient, and one or more sub-bands of non-zero frequency AC coefficients. The number of sub-bands will depend on the size of the block. In the case of a four by four block, only one sub-band is defined. For larger block sizes, a larger number of sub-bands are defined, with separate sub-bands  
10       for the horizontal, vertical, and diagonal high frequency components.

Figure 8 schematically illustrates how the sub-bands are defined for block sizes of four by four, eight by eight, and sixteen by sixteen. For each block size there is a single DC coefficient 810. As depicted, the AC coefficients relate to progressively higher frequency components on moving from the top to  
15       the bottom of the block (higher vertical spatial frequencies), or from the left to the right of the block (higher horizontal spatial frequencies). For a four by four block, the remaining AC coefficients are processed as one sub-band 820.

For an eight by eight block, three additional sub-bands 830, 840, and 850 are defined. Sub-band 830 comprises a four by two group of coefficients of  
20       higher vertical spatial frequency, but lower horizontal spatial frequency, and is immediately below sub-band 820. Sub-band 840 comprises a four by two group of coefficients of higher horizontal spatial frequency, but lower vertical spatial frequency, and is immediately to the right of sub-band 820. The remaining coefficients of an eight by eight block define sub-band 850.

25           For a sixteen by sixteen block, a further three sub-bands 860, 870, and 880 are defined, in addition to those defined for the eight by eight block. Sub-band 860 comprises an eight by four group of coefficients of higher vertical spatial frequency, but lower horizontal spatial frequency, and is immediately below sub-band 830. Sub-band 870 comprises an eight by four group of  
30       coefficients of higher horizontal spatial frequency, but lower vertical spatial

frequency, and is immediately to the right of sub-band 840. The remaining coefficients of a sixteen by sixteen block define sub-band 880.

In some encoding modes, suitable for very low bit rate environments, the AC coefficients may be completely neglected, and only the DC coefficients  
5 processed and transmitted. Such a DC-only mode can still provide useful image data, and offers maximum resilience since (as is described in further detail below) all DC coefficients are fixed length after quantisation. A DC-only mode may be selected, for example, when a fixed size is selected for either the overall image or for selected individual image portions, and it is apparent that  
10 the remaining capacity within the fixed limit is insufficient to allow encoding of the AC coefficients.

In the present embodiment, prediction is performed at the slice level. Thus, each slice includes one reference block and a number of predictions. The predictions are determined in a different manner for the DC and AC  
15 coefficients, as is described below.

#### 2.9.1 DC Prediction

DC prediction in the present example is performed for blocks within a single slice. Figure 9 illustrates the DC coefficients and first sub-band for blocks 912, 914, 916, and 918. The DC coefficient for the first block in the slice is taken  
20 as a reference coefficient. Each subsequent DC coefficient is predicted, from the reference coefficient, as the difference between the current DC coefficient and the preceding DC coefficient. Thus, as shown in Figure 10, the DC coefficients in blocks 912, 914, 916, and 918 are:

783    774    761    729

25 and the prediction process, as illustrated at 1010 accordingly compacts these values to

783    -9    -13    -32

As will be seen, the actual values of the coefficients can then be recomputed at the decoder by adding the predictions -9, -13, and -32 successively to the  
30 reference coefficient.

It is possible (and typical) to predict DC coefficients across a larger number of blocks. For example, in many known image codecs DC coefficients are predicted from a single reference coefficient for a whole image. However it will be seen that a single error in the transmission of the DC coefficients would then propagate across the whole remainder of the image. By constraining the prediction to a smaller number of blocks some compaction is retained whilst confining any errors to a single slice. The consequences of an error within a single slice are further reduced by the checkerboard structure of each slice, since error concealment algorithms used in the decoding process are able to make use of neighbouring blocks if a transmission error results in complete loss of a slice.

### 2.9.2 AC Prediction

Prediction of the AC coefficients is performed for blocks within a single slice, and at the sub-band level. The coefficients in each sub-band are first vectorised. The vectorisation process converts the two dimensional block of coefficients into a string of coefficients. The coefficients are placed in the string in a predefined order. The scanning pattern is arranged such that the resulting string of coefficients is, broadly speaking, ordered from low frequency components to high frequency components. Typically the coefficients are highly correlated when ordered from low frequency to high frequency. A number of scanning patterns can be defined to capture the coefficients in a suitable order. In the present example a zig-zag scanning pattern, illustrated in Figure 8 for each of the sub-bands in exemplary block sizes four by four, eight by eight, and sixteen by sixteen, is used.

By way of specific example, for the four by four block 912 illustrated in Figure 9, the zig-zag order is illustrated by an arrow. In that case the fifteen AC coefficients would be ordered in a vector (as illustrated at 1110 in Figure 11 described below):

-9, -15, -2, -2, 5, -4, 7, -6, -4, -4, 1, 7, 3, -3, -3

A vectorised sub-band for a particular block is predicted on the basis of the sub-band from the previous block, making use of the likely similarity

between the two sub-bands. The prediction method used in the present example follows the method disclosed by Valin and Terriberry in 'Perceptual Vector Quantization for Video Coding', available at <https://arxiv.org/pdf/1602.05209.pdf>. Briefly, the vector spaced is transformed using a Householder reflection. If  $\mathbf{x}_{n-1}$  is a vector defined by the AC coefficients, ordered as above, from either the reference sub-band or the previous sub-band, then the Householder reflection is defined by a vector  $\mathbf{v}$  normal to the reflection plane:

$$\mathbf{v} = \frac{\mathbf{x}_{n-1}}{\|\mathbf{x}_{n-1}\|_2} + s\mathbf{e}_m$$

where  $\mathbf{e}_m$  is a unit vector along axis  $m$  and  $s$  is the sign of the  $m$ th element in  $\mathbf{x}_{n-1}$ . Axis  $m$  is selected as the largest component in  $\mathbf{x}_{n-1}$  to minimise numerical error. The input vector  $\mathbf{x}_n$  is reflected using  $\mathbf{v}$  as follows:

$$\mathbf{z} = \mathbf{x}_n - 2 \frac{\mathbf{v}^T \mathbf{x}_n}{\mathbf{v}^T \mathbf{v}} \mathbf{v}$$

The prediction step describes how well the reflected input vector  $\mathbf{z}$  matches the reflected  $\mathbf{x}_{n-1}$ , which, once transformed, lies along axis  $m$ . An angle  $\theta$  can be calculated to describe how well  $\mathbf{x}_n$  matches the prediction. It is calculated as:

$$\cos \theta = \frac{\mathbf{x}_n^T \mathbf{r}}{\|\mathbf{x}_n\|_2 \|\mathbf{r}\|_2}$$

in which  $\mathbf{r}$  is the vector of prediction coefficients. For the first prediction in a slice,  $\mathbf{r}$  is equivalent to  $\mathbf{x}_{n-1}$ . For decoding,  $\mathbf{z}$  is recovered using the following formulation:

$$\mathbf{z} = \gamma(\cos \theta \mathbf{e}_m + \sin \theta \mathbf{u})$$

where  $\gamma$  is the gain ( $L_2$ -norm) of  $\mathbf{x}_n$ , and  $\mathbf{u}$  is a unit length vector relating the reflected  $\mathbf{x}_n$  to axis  $m$ . The quantities  $\gamma$ ,  $\theta$ , and  $\mathbf{u}$  are subsequently quantised and encoded for transmission as described below. The above operations can then be reversed to recover  $\mathbf{x}_n$  from  $\mathbf{z}$ , and the block reconstructed using the

5 ordering defined by the zig-zag scanning process, which is known to the decoder. In the context of vector quantisation, described further below, this prediction technique has the benefit that resolution is increased around the values in the previous sub-band. In other words, the quantisation scheme will quantise those vectors close to the preceding sub-band vector coarsely than  
10 those vectors lying further away from the preceding sub-band.

## 2.10 Quantisation

Different quantisation approaches are applied for the DC and AC coefficients obtained from the DCT algorithm.

### 2.10.1 DC quantisation

15 For the zero frequency DC coefficients, a fixed-length binary string is used for both reference and prediction coefficients is used. The string can be made longer for a finer quantisation level, or shorter for a more course quantisation level. The length of the string can vary between different image portions to enable different image portions to be encoded at different resolution  
20 levels. Each coefficient is represented by a binary string of the same length, and it is possible for a shorter length to be used for the prediction coefficients than for the reference coefficients. In the present example, as illustrated in Figure 10 at 1030, a seven bit fixed length string is used for both reference and prediction coefficients. This enhances the robustness for the algorithm since  
25 the fixed length string supports resynchronisation if an error occurs during transmission of the encoded image.

### 2.10.2 AC Quantisation

As described above, the AC coefficients are captured in vectors. Vector quantisation techniques are therefore appropriate for quantisation of the AC

coefficients. More particularly, in the present example, gain shape vector quantisation (GSVQ) is used to quantise  $\mathbf{z}$  as defined in 2.8.2 above. GSVQ works by separating a vector into a length and a direction. The gain (length) is a scalar that represents the energy in the vector, while the shape (direction) is a unit-norm vector which represents how that energy is distributed into the vector.

The scalar gain value  $\gamma$  is quantised using a uniform quantiser. For predicted sub-bands, the angle  $\theta$  is also quantised using a uniform quantiser. The shape, or direction,  $\mathbf{u}$ , is quantised using a codebook having  $L - 1$  dimensions and being parametrised by an integer  $K$ . Here  $L$  is the number of AC coefficients in the relevant sub-band. The codebook is created by listing all vectors having integer components which sum to  $K$ . It will be understood that the number of dimensions can be  $L - 1$  because the sum of the components is known. Each vector in the codebook is normalised to unit length to ensure that any value of  $K$  is valid. As the number of dimensions is fixed, the parameter  $K$  determines the quantisation resolution of the codebook. Notably, as  $\gamma$  increases, the quantisation resolution for a fixed  $K$  will decrease. In some examples, therefore, it may be possible to adapt  $K$  based on the computed  $\gamma$ , increasing with increasing gain so as to retain quantisation resolution.  $K$  can also vary between different image portions, as with the DC quantisation, to enable different image portions to be encoded with different resolutions.

An example of this process is illustrated in Figure 11. Block 912 (illustrated in Figure 9) is converted to a vector 1110 using the zig-zag scan described above. After quantisation, the vector 1110 is transformed to a vector 1120. Vector 1120 has the same number of coefficients as vector 1110, but its coefficients are lower valued, and many are quantised to zero. Subsequently, the coefficients are binarised using a fixed length coding method, resulting in the string 1130, with three bits encoding each coefficient. Blocks 914 to 918 are similarly processed, with an additional prediction step performed on the basis of the previous block. The use of fixed length coding at this stage facilitates

resynchronisation at the decoder in the event of errors occurring in transmission.

## 2.11 Coding

Processing as described above results in a series of strings of binary data. For each slice of four blocks, one string represents the DC coefficients, predicted and quantised as described above. There are additional strings for each sub band stack in the slice, the sub band stack being the sub band coefficients for each block in the slice concatenated together. Thus one string represents each of the sub-band stacks of AC coefficients, predicted and quantised as described above, and concatenated for the blocks each slice. Where there are three sub-bands in a block, therefore, each slice will contain three sub-band stacks which will accordingly be represented by three strings. Where there is only one sub-band in a block, each slice will contain only one sub-band stack which will accordingly be represented by one string. Each sub-band stack of AC coefficients is further encoded using a binary arithmetic coding technique. For example, M-coder, disclosed by D. Marpe in "A Fast Renormalization Technique for H.264/MPEG4-AVC Arithmetic Coding," in 51st Internationale Wissenschaftliches Kolloquium, Ilmenau, 2006 can be used. Various other known binary arithmetic coding techniques may also be used.

The binary arithmetic coding scheme makes use of a probability model, which provides for more likely syntax elements in the various strings to be assigned shorter codewords. In the present example, since it can be assumed that the type of imagery to be encoded is known, representative data is used to tailor the probability model. This results in higher compression ratios than would be obtained were the prior known (contextual) approach to be used. Because the sum of each vector of AC coefficients is restricted to be  $K$  as a result of the vector quantisation process, the range of possible values is restricted. Based on a weak invocation of the central limit theorem, the probability of each element is then approximated by a truncated normal distribution with zero mean:

$$p(x_i) \sim \mathcal{N}_{-K,K}(\mathbf{0}, \sigma)$$

The present inventors have identified that the variance  $\sigma$  of this distribution is influenced by vector length  $L$ , vector sum  $K$  and the position in the vector  $i$ . A three-parameter model is used to capture this relationship:

$$\sigma = \alpha \left( \frac{K}{L-i+1} - \frac{K}{L} \right) + K\sigma_0 + \beta$$

5 Parameters  $\hat{\alpha}$ ,  $\hat{\beta}$ ,  $\hat{\sigma}_0$  were estimated using a least-squares optimiser. In

the present example, values for these parameters were computed for medium wave infrared imagery. By way of example, for the first sub-band within a reference four by four block, the values were calculated to be  $\hat{\alpha} = 1.0987$ ,  $\hat{\beta} = 0.8871$ ,  $\hat{\sigma}_0 = 0.0695$ . These coefficients are calculated

10 separately for reference and predictions sub-bands. As will be appreciated, each imaging modality will have a different set of optimal values for these parameters.

The variable length coding scheme is further modified using a bit stuffing scheme, as disclosed by H. Morita, "Design and Analysis of Synchronizable Error-Resilient Arithmetic Codes," in GLOBECOM, 2009. In broad terms, the  
15 scheme allows only  $k-1$  consecutive 1s in the bit stream during encoding. If this rule is breached, a 0 is inserted. An End of Slice (EOS) word of  $k$  1s is used to denote the end of the slice for each sub-band. This ensures the EOS code only occur at the end of the slice in nominal conditions with no errors. The  
20 bit-stuffing scheme further enhances robustness of the coding as it facilitates resynchronisation in the event of an error during transmission.

The variable length coding compresses the information required to code the AC coefficients in each sub-band, but results in strings of different length for each sub-band. This can lead to a loss of synchronisation when decoding in  
25 the event of a bit error occurring as a result of transmission. Whilst it is possible to add further codewords to enable resynchronisation, there remains the problem that if these words are corrupted, there will remain the potential to lose synchronisation with consequential and potentially significant errors arising in the decoded image. In the present example, therefore, an allocation method is

used to convert the variable length bins into bins of consistent length. The allocation method used in the present example is based on the error-resilient entropy code (EREC) method. This method is disclosed by D. W. Redmill and N. G. Kingsbury in "The EREC: an error-resilient technique for coding variable length blocks of data", IEEE Transactions on Image Processing, vol. 5, no. 4, pp 565-574, April 1996; a faster method being disclosed by R. Chandramouli, N. Rangahathan, and S. J. Ramados in "Adaptive quantization and fast error-resilient entropy coding for image transmission", IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 4, pp 411-421, August 1998.

In broad terms, the EREC methods in the above-referenced disclosures apply a bin packing scheme to convert bins of variable length for each block in an image into fixed length bins, moving bits from relatively longer bins into relatively shorter bins. A defined search strategy is used to identify the relatively shorter bins so that they are filled in an order that can be known to the decoder. Since the bin length can be fixed, it does not need to be provided to the decoder in transmission, and there is no need to include further synchronisation words. The decoder can unpack the fixed length bins using knowledge of the search strategy, bin length, and an end of block code. The EREC has the additional benefit that errors are more likely to occur at higher spatial frequencies, where errors are likely to be easier to detect and conceal.

As schematically illustrated in Figure 12, an exemplary slice 1210 of four eight by eight blocks comprises four DC coefficients, and four sub-bands of AC coefficients for each block. As described above, the DC coefficients are of fixed length as schematically illustrated at 1220. The AC coefficients for each sub-band are illustrated schematically as being stacked at 1230, 1240. Because of the variable length coding, the strings for the AC coefficients in each sub-band stack are of variable length. An allocation method based on the EREC method is applied to each sub-band stack in each slice, resulting in uniform length bins (such as 1250, 1260) for each sub-band in the slice. In contrast the EREC method in the above disclosures is applied at the block level.

Sub-band stack 1230 comprises strings representing the lower frequency AC coefficients for each of the blocks in the slice 1210, arising from sub-bands

1211 in a first block, 1212 in a second block, 1213 in a third block, and 1214 in a fourth block. After variable length coding there are four strings, 1231, 1232, 1233, and 1234 respectively, representing these coefficients, each string having a different number of bits. Four bins of uniform size are defined, such that the  
5 four bins in combination have sufficient space to hold all the bits of strings 1231, 1232, 1233, and 1234. The bits representing the sub-band stack are then allocated to the bins. Each bin has an associated block, and the bits in that bin start with the bits for the relevant sub-band of its associated block. If the number of bits in the relevant sub-band of its associated block is greater than  
10 the uniform size, the allocation method interrogates the next bins sequentially to determine if there is space for the excess bits.

For example, in the case of the bits representing the sub-band stack comprising the sub-bands for the lower frequency AC coefficients in a slice, if the string 1231, which represents the coefficients of the lower frequency AC  
15 coefficients in the first block, is longer than the uniform size, the allocation will first interrogate the bin associated with the second block to determine if there is space for the excess bits. If there is space, the excess bits are placed in that bin. This step is repeated for each of the strings 1231, 1232, 1233, and 1234. Thus, if there are excess bits in string 1232, the allocation method will  
20 interrogate the bin associated with the third block, and so on for strings 1233 and 1234 (the bin associated with the first block being interrogated in the case that there are excess bits in string 1234). If there remain excess bits once this step has completed, the allocation method repeats the step, but instead of interrogating the bin associated with the subsequent block, it interrogates the  
25 bin associated with the next-but-one block. The step is repeated, interrogating sequentially later blocks, until all the bits are allocated to one of the bins.

The bins are thus filled firstly with bits representing the relevant sub-band of their associated blocks, and then, in a sequential order, excess bits from the relevant sub-bands of other blocks in the slice. The decoder can unpack the  
30 fixed length bins using knowledge of the search strategy, bin length, and the EOS word of  $k$  1s that is inserted at the end of each sub-band in the slice.

It will be noted that the bins for different sub-bands may have different (fixed) lengths. This implementation enables the length of the slice to become a parameter, which may for example be defined for each image portion, and which enables resilience to be traded with bandwidth and computation time.

5 Smaller length slices are more resilient, but require larger bandwidth, and longer computation times.

In some examples, early stopping criteria are applied to the EREC bin packing. Because of the complexity of the processing, it is possible for many iterations to be run without finding an exact packing. By terminating after a  
10 certain number of iterations, in both bin-packing during encoding and unpacking during decoding, a suitable packing (or unpacking) can be arrived at, without significant error, whilst ensuring that the processing terminates within a reasonable time.

A bit stream is then created by concatenating the uniform length bins in  
15 successive steps, as is illustrated in Figure 13. In a first step, the uniform length bins for each sub band are conceptually flattened together, resulting in separate strings 1310, 1320, 1330, 1340, and 1350 for the DC coefficients, and for each sub-band in a slice. The separate strings are then concatenated for each slice, resulting in a string 1360 containing all the information for one slice. All slices  
20 within a set of blocks are combined as illustrated at 1370, and then the sets for an image portion are combined as illustrated at 1380. The concatenation steps are performed in order to preserve the identity of each set, slice, sub-band and block. The image portion header, including the size of the image portion in terms of number of bits, is added to the concatenated bitstream for the image  
25 portion. Each of the image portions is processed as described. The image portions can then be interleaved, encrypted, and transmitted independently, or, as in the present example, the bitstreams for each of the image portions are concatenated, and the resulting bitstream, which encodes the whole of the image, is interleaved and encrypted, as illustrated schematically at 1390 and  
30 described below.

## 2.12 Interleave

Prior to transmission, the binary stream from the encoder is split into data packets. Whilst this can be done by simply splitting the stream into components of the appropriate packet size, in the present embodiment an interleaver is used. The interleaver selects which bits are integrated into which data packet. This can be done in an ordered manner, or at random, with a key provided to the decoder so that the received packets can be re-ordered so as to be able to re-construct the image. The interleaver has the effect that, should packet losses or burst errors occur, the errors in the re-ordered binary stream will be distributed throughout the image, rather than concentrated in any one area. Distributed errors can be easier to conceal. In the present embodiment, as described above, the bitstreams created for each of the image portions are concatenated together prior to interleaving, so that any errors are distributed across the entire image, rather than across only one image portion.

In the present embodiment a block interleaver is used. The block interleaver writes data into a section of allocated memory row-wise, and then reads data from the memory column-wise into packets for transmission. This distributes neighbouring bits across different packets. The number of rows in the allocated memory is selected to be the size of the data packet. The number of columns is selected to be the maximum number of packets required to encode an entire image. This is illustrated in Figure 14, in which the memory allocation is shown schematically with cells 1410 containing data, and cells 1420 that do not. After interleaving, therefore, some data packets contain null information at the end of the binary stream. The null information enhances the resilience of the encoded data, since it has no effect on performance if it becomes corrupted. In alternative embodiments it may be possible to base the memory allocation size on the actual encoded length, rather than using a fixed size, thereby reducing the amount of data required to encode an image.

As each data packet is read from the block interleaver, it is assigned a header containing the frame number and packet number. The packet can then be transmitted.

Typically interleaving is performed in a separate dedicated transmission apparatus. In the present example, however, the interleaving is performed as an integral part of the encoding of the image. As has been described above, where it is desired to encrypt the image, the encryption can also be performed  
5 as an integral part of the encoding of the image. Thus, the encoded image file can, for example, be stored on a computer-readable medium after the interleaving and encryption has been performed. The encoded image file can be passed to a separate dedicated transmission apparatus, with interleaving and encryption already performed, so that the dedicated transmission apparatus  
10 need not perform either interleaving or encryption prior to transmitting the image. This has the benefit that the dedicated transmission apparatus does not need to read through the encoded image data in order to perform interleaving or encryption prior to transmitting the data packets. Thus including the interleaving step in the encoding process reduces latency in transmission, as well as  
15 providing additional resilience, particularly to burst errors.

## 2.13 Encryption

In some circumstances it can be desirable for the image data to be encrypted prior to transmission. Encryption reduces the potential for unauthorised third parties to gain access to the image during transmission. A  
20 number of encryption methods are well-known and can be used in the present method. One example method is the Advanced Encryption Standard (AES). In the present example AES 256 is used and applied after interleaving. The use of encryption reduces the resilience of the encoded image to data loss, because the loss of only one bit from an array of 16 encrypted bytes results in the entire  
25 16 byte array being unrecoverable. However the enhanced resilience resulting from other steps applied in the codec still ensures that images encoded using the present method are significantly more resilient than those encoded using currently known image codecs, even when encryption is applied.

The encryption algorithm is included as an integral part of the image  
30 encoding. Where AES 256 is applied, the size of the data packets is selected to be a multiple of the encryption array size of 16 bytes. Because the packet size is chosen to be optimal for the encryption, it is ensured that no additional

padding is required. Were, for example, the packet size to be selected to be 300 bytes, additional padding with null information would be required to make the packet size divisible by 16 bytes, since this is a requirement of AES 256. Thus encryption efficiency is increased.

## 5 2.14 Decoder

The transmitted packets are received at a decoder and, in broad terms, can be processed by reversing the steps described above so as to reconstruct the image transmitted, as described above in Section 1 with reference to Figure 1b. Some steps are taken by the decoder to increase robustness, and some  
10 steps are taken to identify errors that may have occurred during transmission. Some steps are also taken by the decoder to conceal any errors that have occurred.

When a packet is received at the decoder, and subsequent to any necessary decryption, the frame number and packet number are read from the  
15 header. If the frame number is greater than then previous packet, then the binary stream is read out of the block interleaver and the decoder runs to produce an image. If the frame number is the same as the previous packet, then the packet number is read and the payload contained within the packet is written into the block interleaver based on the packet number. If the frame  
20 number is less than the previous packet, then the packet is discarded.

The decoder is able to decode the binary stream using parameters provided to it in the image header and other predetermined factors that can be pre-programmed in the decoder, such as the inverse of the allocation method used to allocate bits to a position in the bit stream. Since the binary stream  
25 arises from a number of strings that are concatenated together in a predetermined order, each slice, and the bits representing its zero frequency coefficients for each block, and the sub band stacks, can be identified from the binary stream. Within the slice, the position of bits for particular sub-bands of particular blocks is determined by the allocation method described above.

30 Similarly, separate image portions can be identified from the image header, and individual image portion headers. The separate image portions

can be decoded independently, for example using a multithreaded implementation, similar to the implementation described above in relation to the encoding method.

5 The decoder can identify a bit stream relating to each sub band in each block by locating the end of slice code word and inverting the steps of the allocation method. For example, to separate the bits relating to a sub-band of each block in a slice, the decoder first identifies each of the bins in the bit stream for the slice. Each bin has an associated block in the slice. The decoder can then read the start of each bin in the slice. If the decoder reads an  
10 end of slice code word a bin, the bits read to that point relate to the complete sub band for the block associated with that bin.

If it does not read an end of slice code word in a first of the bins, which first of the bins is associated with a first block, it moves to read a second, subsequent bin to determine whether an end of slice word is present. If it  
15 identifies an end of slice code word in the second bin, the bits immediately following the end of slice code word relate to the remainder of the bits for the sub band of the first block. If it identifies a further end slice code word in the second bin, then the bits for the sub band of the first block are complete. If the decoder does not identify a further end of slice code word in the second bin, it  
20 moves to the subsequent, third bin, where it will need to identify two end of slice code words before reading bits relating to the remainder of the sub-band of the first block. This process can repeat until the sub-band is complete for all of the blocks in the slice.

In some examples the predicted DC coefficients can be capped.  
25 Typically, the changes in the actual DC coefficients from block to block will be relatively small, and a relatively large change can be indicative on an error or (in the decoder) corruption occurring during transmission. Imposing a cap on the value of the predicted DC coefficients, constraining their values to be within a certain range, can reduce the resulting errors in the decoded image.

30 In the exemplary case above for blocks 912, 914, 916, and 918, a fixed cap for the predicted coefficients of  $\pm 50$  may be appropriate. Such a fixed cap, in this example, would not affect the true values of the predicted coefficients but

would remove large errors that may occur in transmission. For example, were the predicted coefficient -9 for block 914 to be corrupted to -999, in the absence of a cap, an error of 990 would be present in block 914, and would propagate through blocks 916 and 918. A fixed cap of  $\pm 50$  would reduce the error to 41,  
5 and result in a lower amount of distortion in the final decoded image.

The cap need not be fixed. For example, the cap may vary dynamically between blocks, slices, or image portions. It may be defined as a percentage of the reference block DC coefficient; or alternatively as a percentage of the reference block DC coefficient but with a set minimum value. A set minimum  
10 value avoids the potential for a percentage cap to be too small if the reference DC coefficient is small. In an alternative to the use of capping, in some examples it may be appropriate for the decoder to reject values for DC coefficients that fall outside a certain range. Pixels with rejected DC values can be left blank; replaced with an estimate based on neighbouring pixel values; or,  
15 if the image is part of a series of frames in a video, replaced with the corresponding value from the previous frame.

The decoder implements a check to determine that the coefficients of the reconstructed block add up to  $K$ . The decoder can identify the value of  $K$  from the header information. The encoding mode specified in the image header may  
20 specify the value of  $K$ ; or the value of  $K$  may be specified in each image portion header, as would be appropriate if the quantisation level is to vary between image portions. If the coefficients do not add up to  $K$ , as will be understood from the above, it is apparent that an error must have occurred. The error may in some examples be corrected by simply adding or subtracting the appropriate  
25 value from the maximum coefficient so as to ensure that the overall sum does add to  $K$ . Alternatively or in addition the error can then be signalled to an error concealment module of the decoder, described below.

Once the coefficients have been determined by the decoder, the image data can be reconstructed by performing an inverse of the discrete cosine  
30 transform described above, and applying a post filter to invert the pre-filter described above.

In the present example an error concealment method based on a persymmetric structure of optimal Wiener filters is used to conceal any identified errors. This method is able to isolate errors that have been identified and prevent their propagation to neighbouring blocks. Effectively, an identified  
5 corrupted block can be interpolated using the Wiener filter.

Errors can be identified using known methods to detect visual artefacts in the decoded image. Errors can also be identified using information obtained from the decoding process. Such information may include sum-checking during the reconstruction of vectors in the reverse GSVQ process; or from the bit-  
10 stuffing scheme applied during coding. Where the image is part of a series of frames of video footage, it will be possible to use information from a previous frame to replace information lost as a result of transmission errors in the current frame, rather than using the interpolation method above.

### 15 3. PERFORMANCE

The performance of the codec has been analysed for resilience against bit errors occurring during transmission, and compared to the performance of known image codecs with the same bit error rates. Bit error rates have been simulated through the random flipping of bits.

20 Figure 15 is a graph illustrating the variation of decoded image quality, described by peak signal-to-noise-ratio (PSNR), with bit error rate, for an example of the present invention, illustrated by line 1510, and current image codecs JPEG, JPEG2000 (J2K), H.264 and HEVC, illustrated by lines 1520, 1530, 1540, and 1550 respectively. For the current codecs, the versions most  
25 resilient to bit error rate were selected for comparison. It can be seen that the examples of the present invention have lower image quality at low or zero bit error rates than most current codecs, but that image quality is maintained for significantly higher bit error rates than for all current image codecs. All the current image codecs shown suffer catastrophic image loss for bit error rates of  
30  $10^{-3}$ . The HEVC codec shows significant reduction in quality even for bit error rates of  $10^{-6}$ . By contrast, line 1510, illustrating the performance of an example of the present invention, shows almost no reduction in PSNR for bit error rates

of up to  $10^{-3}$ , a relatively slow loss of quality thereafter, and useful information still obtained at a bit error rate of  $10^{-1}$ . In other words, examples of the present invention enable useful image data to be communicated via a transmission channel in which one bit of every ten is lost or corrupted.

5           Figure 16 further illustrates the robustness of an example codec to bit errors. Figure 16 shows a number of actual images, coded using an example codec and decoded after simulated corruption of data. Image 1610 illustrates the image with bit error rate of  $10^{-6}$ . Image 1620 illustrates the image with a bit error rate of  $10^{-5}$ . Image 1630 illustrates the image with a bit error rate of  $10^{-4}$ .  
10          Image 1640 illustrates the image with a bit error rate of  $10^{-3}$ . Image 1650 illustrates the image with a bit error rate of  $10^{-2}$ . Even at the highest bit error rate, useable information can still be obtained, particularly if the image is part of a stream of frames of the same scene. No loss of usefulness is seen at a bit error rate of  $10^{-3}$ , with image problems that may impair algorithms such as  
15          target detection algorithms only occurring at bit error rates of  $10^{-2}$  or higher. It will be noted that current image codecs such as H.264 suffer catastrophic loss at a bit error rate of  $10^{-4}$ .

#### 4. IMPLEMENTATION DETAILS

20           In order to implement the codec, various selections have been made to support rapid processing speed. For example, lookup tables are used to emulate floating point operations such as computation of exponentials or cosines. Integer approximations or integer scaling are used where lookup tables would be unfeasible because of size. Integer approximations are  
25          expected to be most beneficial because they minimise complexity with only a small reduction in precision. This is done, for example, in the implementation of the lapped filter and discrete cosine transform using the lifting process described above. Integer scaling is used for other calculations, such as computation of square roots or the vector norm.

30           A number of fixed and repeatable operations are stored within lookup tables, including quantisation tables, the scanning order of coefficients, lapped filters, fixed length codewords, and DCT parameters. Some operations that

could be stored within lookup tables, such as the probability model for arithmetic coding and the vector quantisation function, are currently computed outside of lookup tables because of the memory requirement, but could be implemented as lookup tables in future implementations.

- 5           As has been noted in the above, the various parameters of the encoding and decoding can be configured for particular applications or uses of the encoding and decoding method. Different parameters can be selected to enhance resilience of the encoded data to transmission errors, or to enhance the amount of compression applied to an input image, or to enhance image
- 10   quality. The configurable parameters can include: the factor by which an input image is downsampled; the number of bits used to encode the AC and DC coefficients (both reference coefficients and predicted coefficients for DC, and both reference and predicted gain and angle  $\theta$  for AC coefficients); maximum values for AC and DC coefficients (reference and predicted); quantisation
- 15   levels, including the parameter  $K$  used for vector quantisation of the AC coefficients; the size of the macroblock; whether or not to operate in a DC only mode; the number of repetitions of the header; whether or not to operate error concealment algorithms such as the Wiener error concealment technique; the number of times the fixed length used for the purposes of EREC is repeated;
- 20   the maximum length of a binary slice; whether or not the blocks are split into sets (such as the interlocking checkerboard sets illustrated in Figure 6); the length of any synchronisation words used; the bit stuffing frequency used during EREC processing; whether or not the transmitted bit stream should be split into uniform size packets, and, if so, what size the packets should be; whether or not
- 25   the overall binary length of the encoded image should be fixed, and if so, to what length (image quality being reduced as necessary to enable the length restriction to be met); and the number of portions into which an image is split.

In one example, the encoding and decoding has been implemented on a reference design board having 1.2GHz CPU Cores, Platform 600MHz and

30   DDR3 memory 800MHz, and containing two e500v2 cores. As noted above, a benefit of processing in independent image portions is that multiple asynchronous cores can be used to significantly increase processing throughput.

Transmission of the encoded image can be performed by a separate, dedicated transmission apparatus. The dedicated transmission apparatus may be a separate processor coupled to an appropriate transmitter and antenna. In some examples, it is expected that the method for encoding an image may be performed on an aerospace platform such as an unmanned air vehicle or a missile. In such a system the encoding may be performed on an image processor receiving images from an imaging sensor, such as a camera operating in the visible or infra-red wavelength bands. Interleaving can be performed on the image processor. An encoded image file can then be passed to a second processor linked to the platform's communication antenna, and the image transmitted to a ground station, or to a remote operator. As described above, performing the interleaving on the first processor reduces latency in the transmission of the images, as well as providing additional resilience, particularly to burst errors.

Figure 17 is a schematic illustration of such an exemplary system. An unmanned air system such as a missile 1 comprises a sensor 2 that is operable to capture images of its field of view. The sensor outputs image data to a first processor 3 which is in communication with a memory 4. The image data may for example comprise a number of pixels, each pixel defining an intensity value for a small component area of the image. For a greyscale image, each pixel need only define one intensity value. The processor 3 operates to encode the image data into a bit stream which may be stored in memory 4 for later transmission, or which can be passed to a dedicated transmission apparatus 5 for wireless transmission to ground station 6. Dedicated transmission apparatus 5 can include both an antenna for transmitting signals and a second processor for controlling the transmission process. Ground station 6 comprises an antenna 7 for receiving communications such as the bit stream encoding the image from unmanned air system 1. The antenna 7 passes received data to a processor 8, which is operable to decode the image. The decoded image may be stored in memory 9. In some examples the decoded image may be processed further by processor 8, for example to track a target through a series of images in a video stream received from unmanned air system. In some examples the decoded image may be displayed to a user for human analysis.

For this purpose a user terminal 100 is provided in communication with the processor 8. In other examples the decoded image may be output to another system for further analysis.

5 Disruption during wireless transmission of the bit stream from the unmanned air system 1 to the ground station 6 can result in errors in the bit stream received at the ground station 6. The impact of these errors on the useability of the image can be mitigated through altering the coding used for the image, for example using the techniques described below.

10 Whilst in the above a number of specific embodiments of the invention have been described, it will be appreciated by those skilled in the art that variations and modifications may be made to the above described embodiments without departing from the scope of the invention, which is defined in the accompanying claims. For example, whilst in the above the use of the encoder  
15 is described with reference to monochromatic images, it will be appreciated that the encoder and decoder can also be applied to colour images, or to multi-spectral images, or to hyper-spectral images. Colour images can be encoded using standard techniques for representing colour in image data in which separate channels are used to represent different colour components of an  
20 image; or by using a YUV-type colour space, in which the Y channel represents a grayscale image comprising a weighted sum of the red, green, and blue components, and the U and V channels represent data obtained by subtracting the Y signal from the blue and red components respectively. Such techniques exploit the correlation between the different colour components that is common  
25 in visible imagery. Similar techniques may also be appropriate for different image modalities.

It will also be appreciated that, whilst in the above it has been described to use image portions in the form of strips, any shape of image portion can be used. For example, the image portions could be in the form of columns; or in  
30 the shape of squares or rectangles.

It will also be appreciated that, whilst a specific order of processing steps has been described in the above, in certain cases it may be possible to vary the order in which the steps are applied. For example, whilst it has been described to apply prediction before quantisation, it will also be possible to apply

quantisation before prediction. It will also be possible to apply downsampling at a different stage of the overall coding process. For example, downsampling may be performed as a first step, before processing of the image into image portions.

5           It will also be appreciated that alternative methods can be used to implement certain processing steps. For example, other frequency based transforms can be used in place of the discrete cosine transform. Such transforms may include for example the discrete sine transform; a discrete wavelet transform, such as for example the Haar, Daubechies, Coiflets,  
10   Symlets, Fejer-Korovkin, discrete Meyer, biorthogonal, or reverse biorthogonal; the discrete Fourier transform; the Walsh-Hadamard transform; the Hilbert transform; or the discrete Hartley transform.

          Whilst, in the above, it has been described to use an objective function that models end-to-end channel distortion when optimising the lapped filter, it  
15   will be understood that other objective functions can also be used in the optimisation process. For example, an objective function can be used to obtain a maximum in an image quality metric. Alternatively, an objective function that relates to distortion in the coded image can be minimised. Such an objective function does not take account of transmission errors, but may be appropriate  
20   where transmission errors are difficult to model or unknown.

          Whilst it has been described in the above to use a block interleaver to distribute the binary stream amongst data packets for transmission, it will also be appreciated that other interleaving methods can be used. For example, a random interleaver can be used. A random interleaver creates an array, in  
25   which each array element contains its own index. The array is then randomly shuffled to produce an array of randomly arranged indexes. When copying the binary stream into the interleaved binary stream, each element of the stream reads the random index assigned in the random array, and is then copied to that index in the interleaved binary stream. When receiving the data, the  
30   opposite is performed. The header information in the transmitted data packets may then contain information determining how to reconstruct the array at the decoder. Such a random interleave process may for example be used to

provide additional security to the data stream during transmission, since a seed used to generate the random array could be stored at the encoder and decoder, and not transmitted. The interleave process may alternatively be omitted.

5 Finally, it should be clearly understood that any feature described above in relation to any one embodiment may be used alone, or in combination with other features described, and may also be used in combination with one or more features of any other of the embodiments, or any combination of any other of the embodiments.

## CLAIMS

1. A method for encoding data defining an image, the method including the step of providing metadata associated with the image, encoding the metadata into binary code to form a metadata string, and repeating the metadata string a number of times.  
5
2. The method of claim 1, further comprising the steps of:
  - segmenting the image into image blocks, each image block in the portion having a uniform block size;
  - 10 - applying a frequency-based transform to each of the image blocks, thereby providing transformed image data in which the image data is represented as coefficients defining a linear combination of predetermined basis functions having different spatial frequencies;
  - quantising the coefficients; and
  - 15 - converting the quantised coefficients into binary code.
3. The method according to claim 2, wherein the metadata string is repeated at least three times.
- 20 4. The method according to claim 2 or claim 3, wherein the metadata string is repeated at least five times, and preferably at least seven times.
5. A method of decoding a bitstream to reconstruct an image, the method comprising the steps of identifying, in the bitstream, a metadata string containing bits relating to metadata associated with the image;  
25 determining the number of times the metadata string is repeated; and, for each bit in the metadata string, applying a voting procedure to determine the value of each said bit.
- 30 6. A method of encoding a series of image frames including at least a current frame and a preceding frame, each of the frames being encoded according to the method of any one of claims 1 to 4.

7. A computer-readable medium having stored thereon data defining an image, which data has been encoded according to the method of any one of claims 1 to 4.
- 5 8. A computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method of any one of claims 1 to 6.
9. A processor configured to perform the method of any one of claims 1 to  
10 6.

1/9

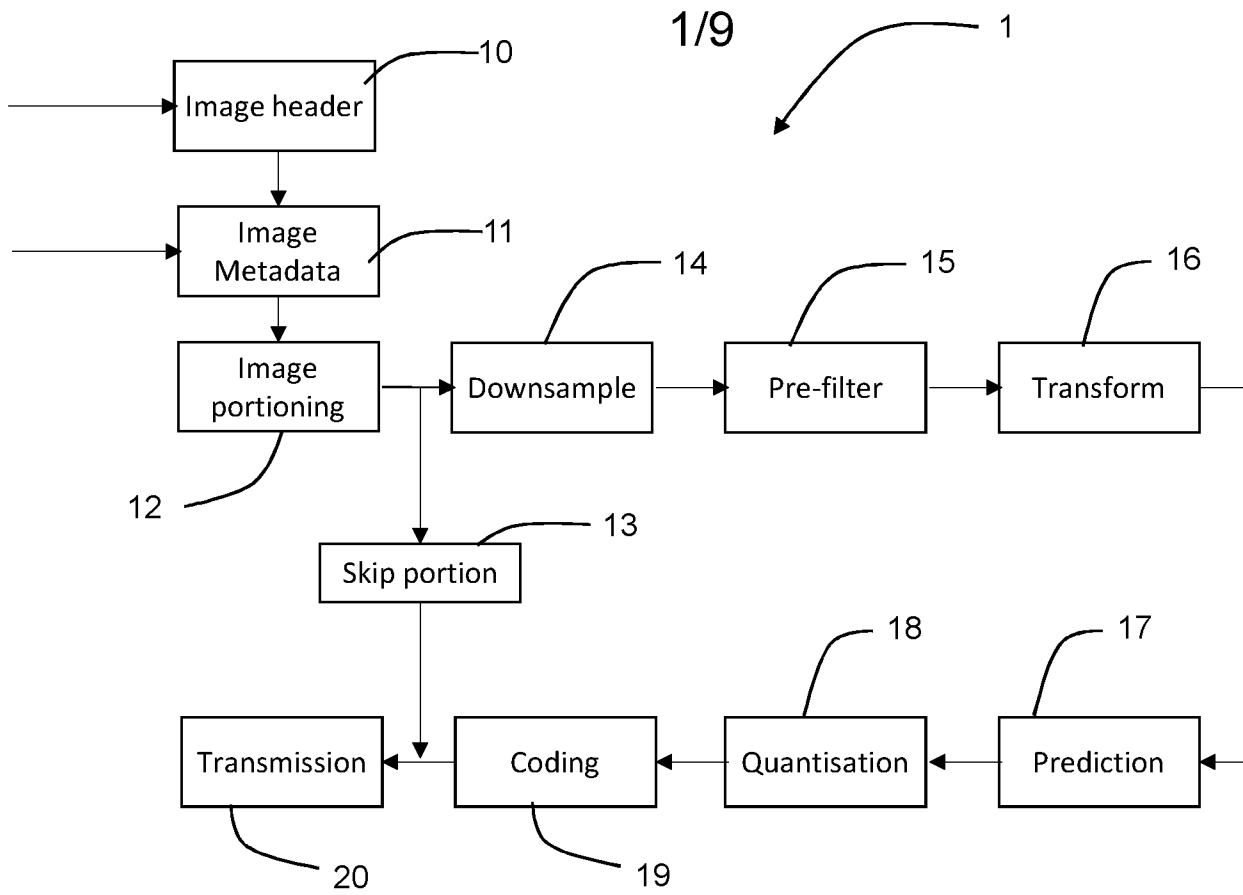


Fig 1A

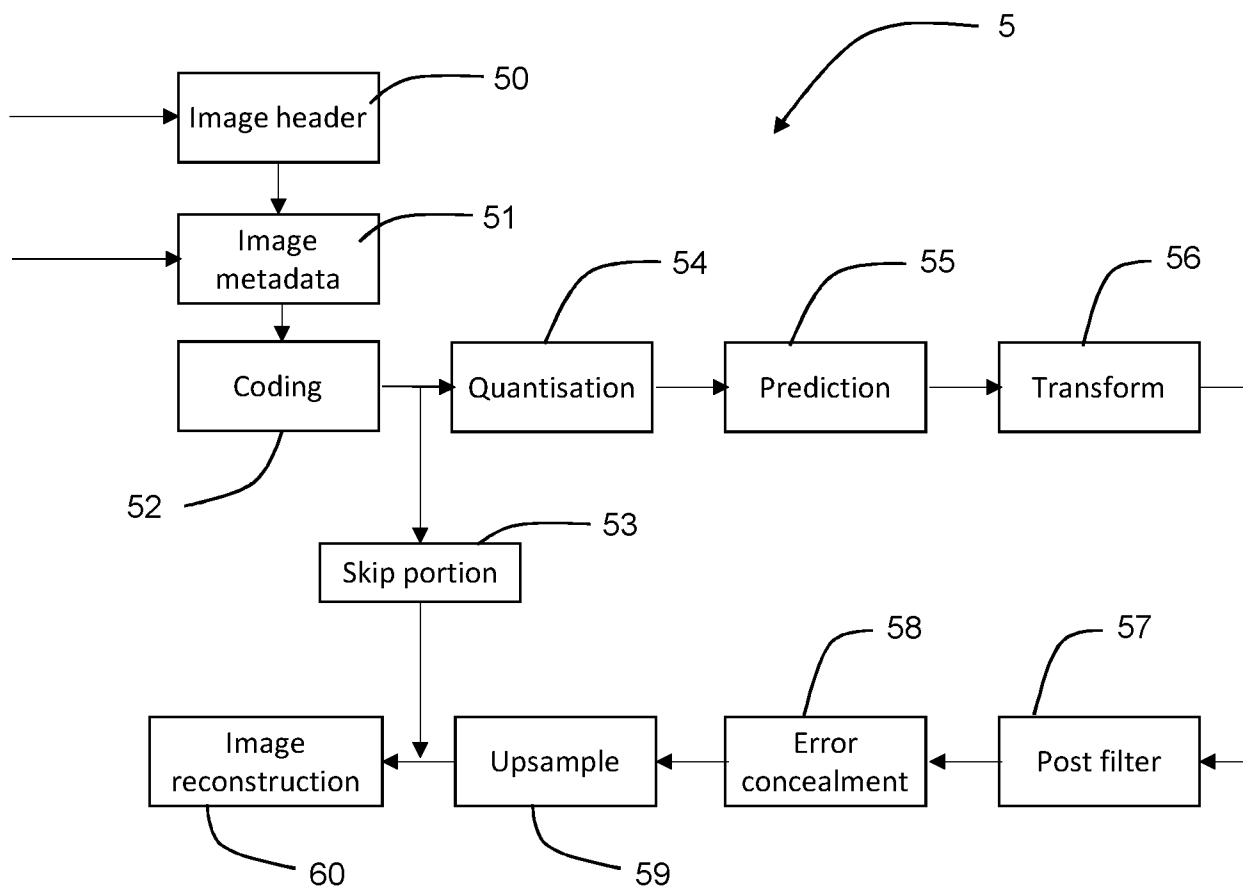
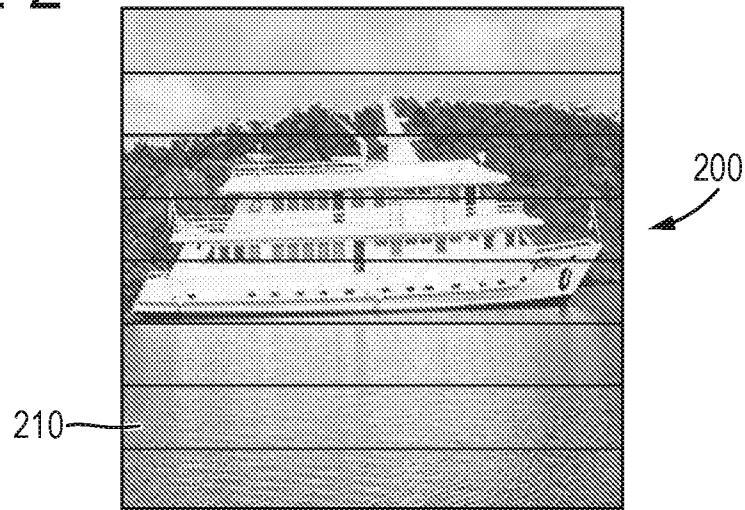


Fig 1B

2/9

Fig. 2

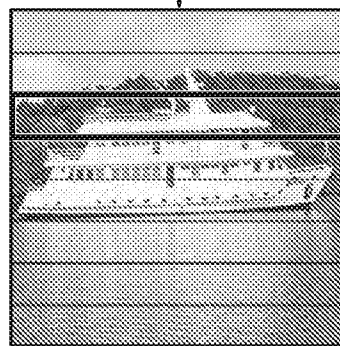
Image portioning



Downsample

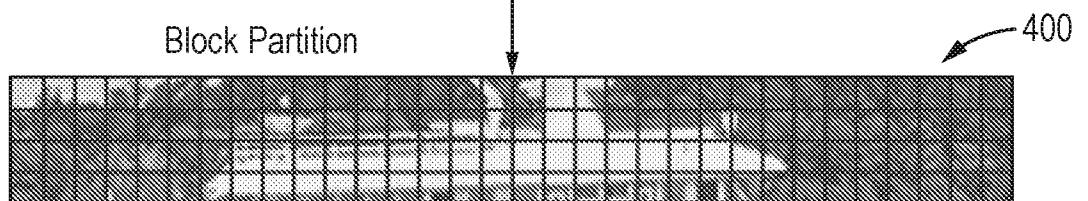
Fig. 3

300



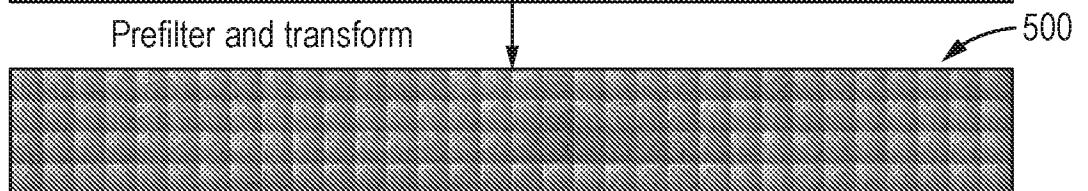
Block Partition

Fig. 4

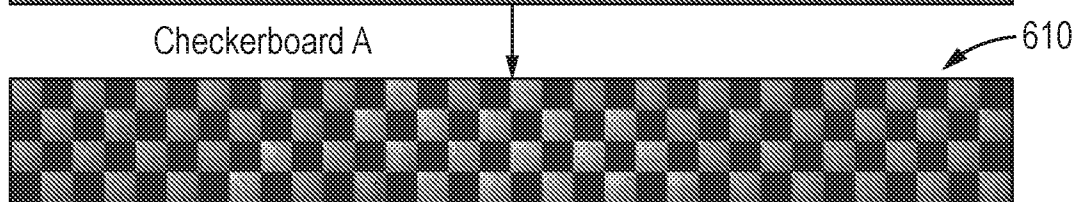


Prefilter and transform

Fig. 5

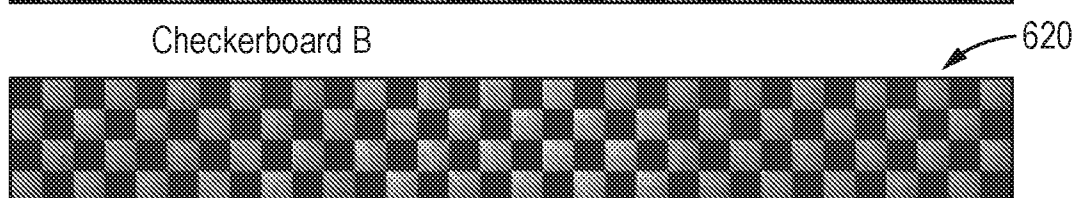


Checkerboard A



Checkerboard B

Fig. 6



3/9

Fig. 7

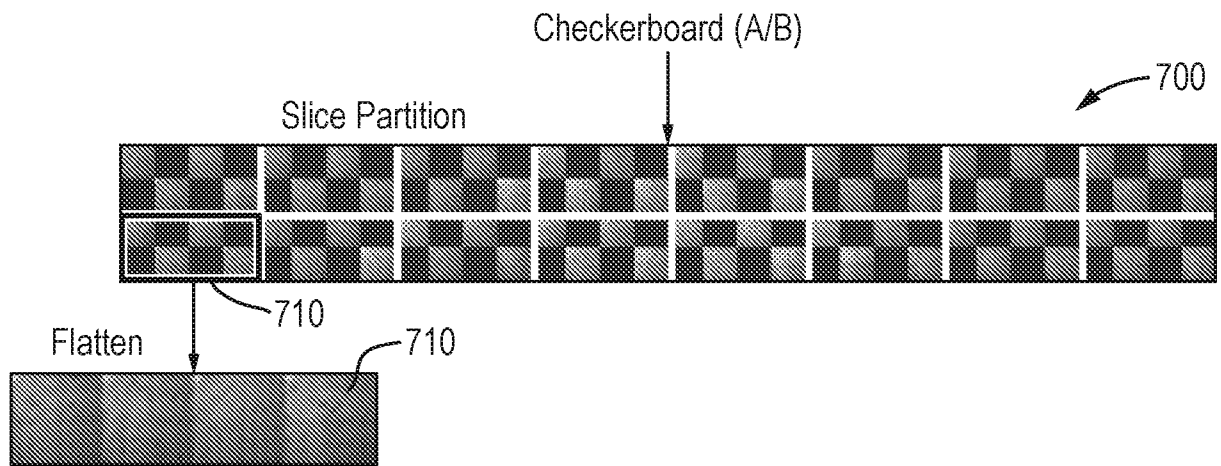


Fig. 8

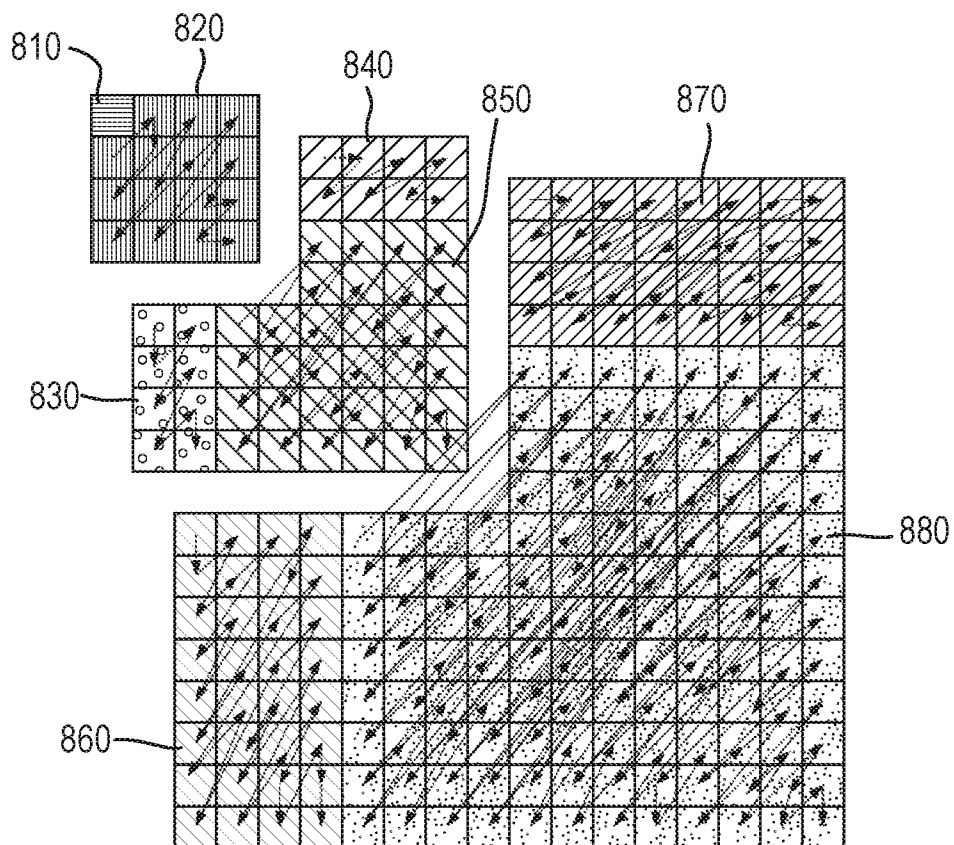


Fig. 9

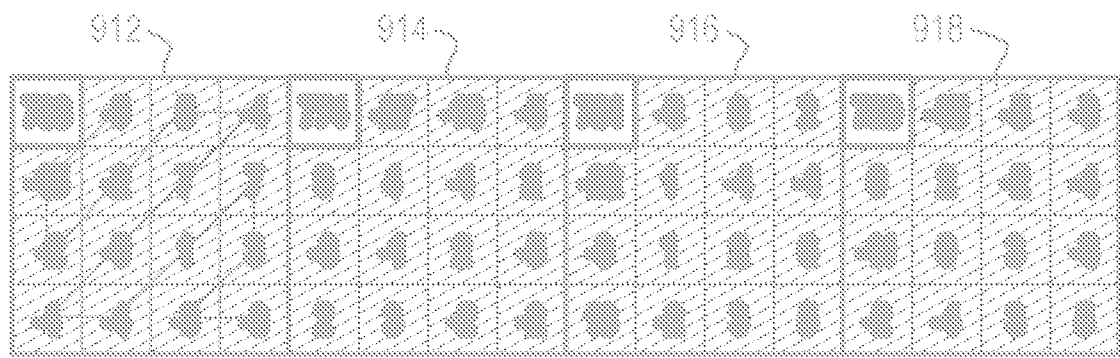


Fig. 10

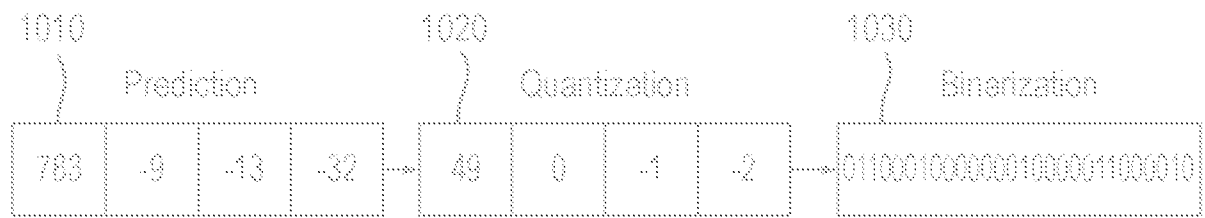
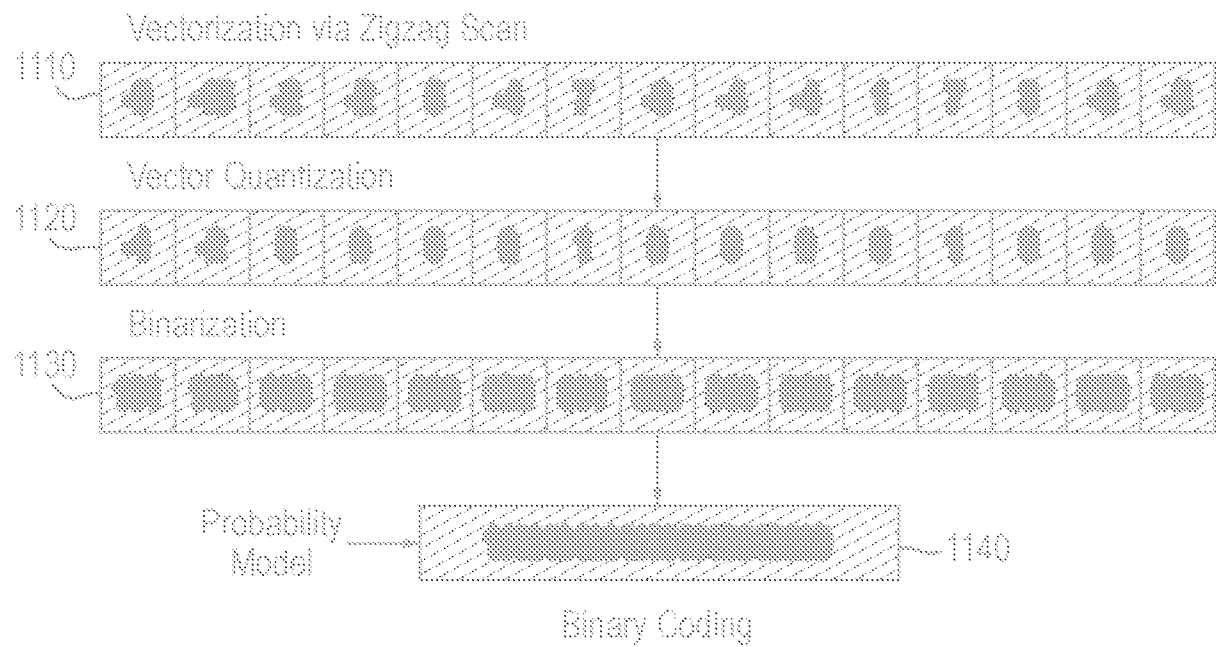


Fig. 11



5/9

Fig. 12

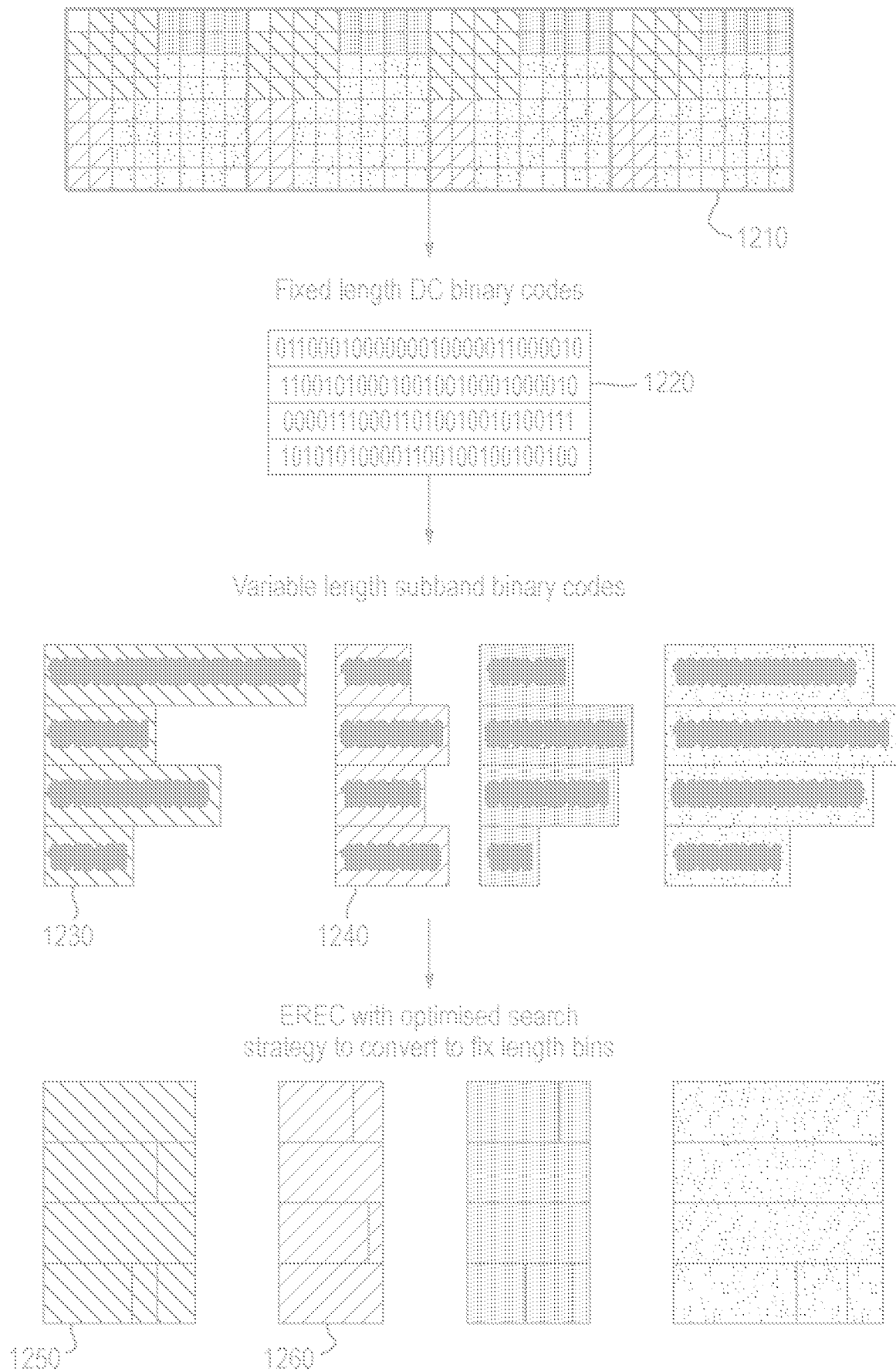


Fig. 13

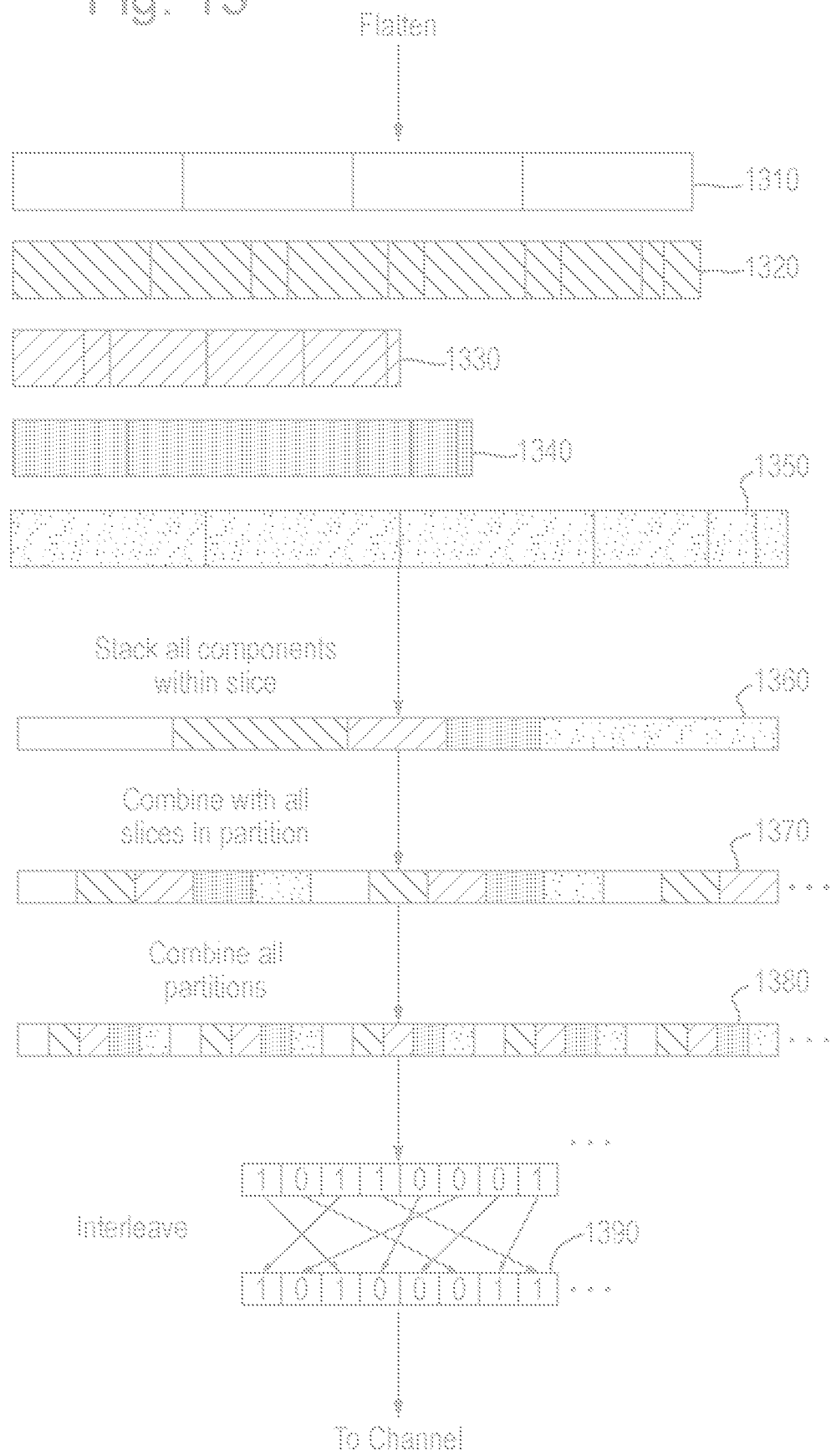


Fig. 14

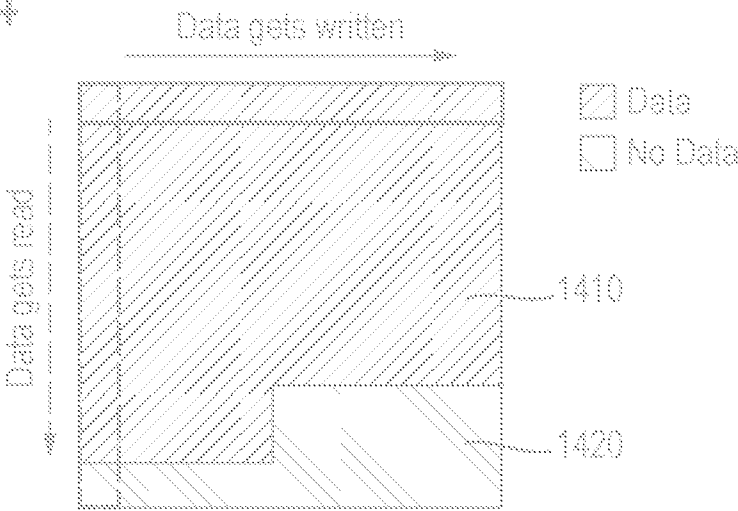


Fig. 15

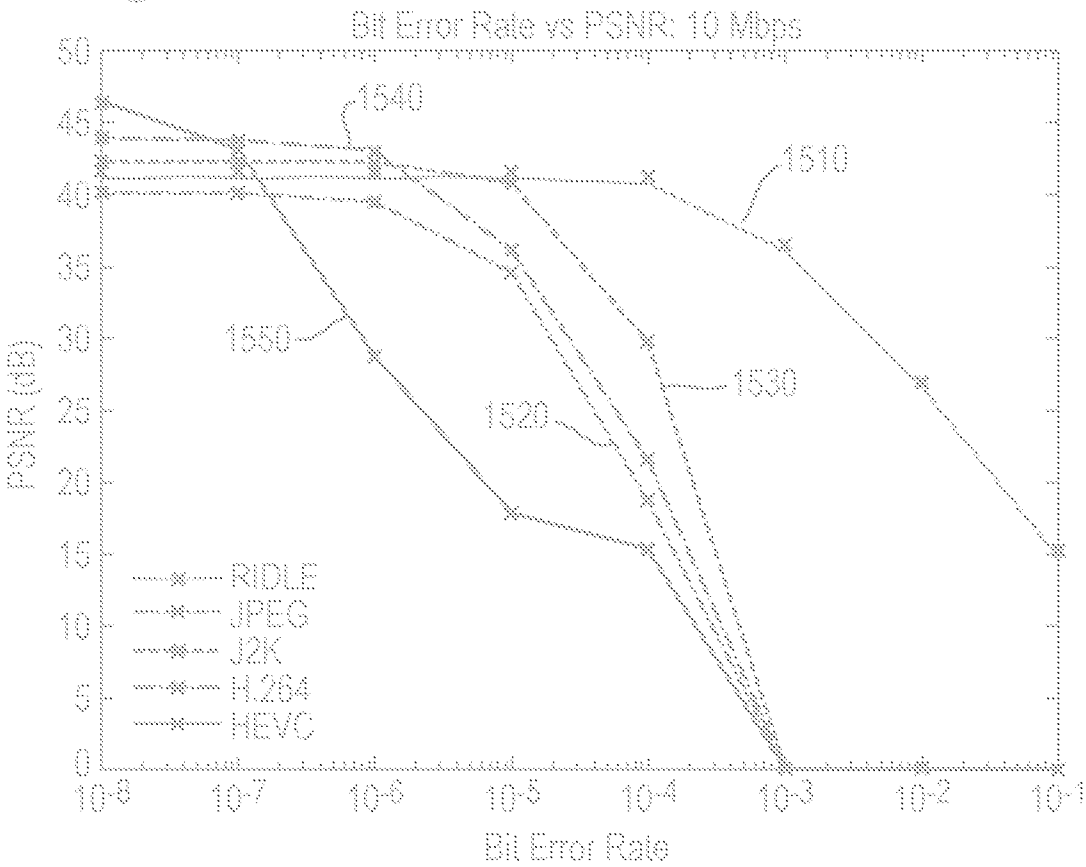
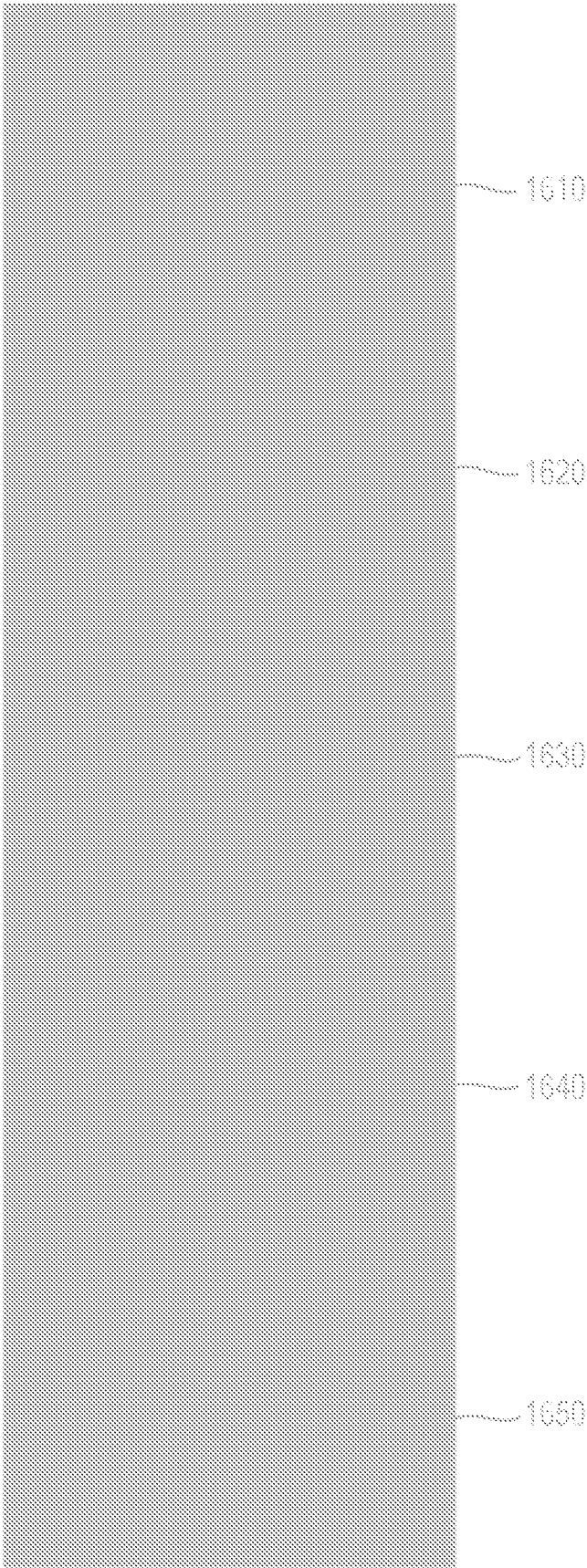


Fig. 16



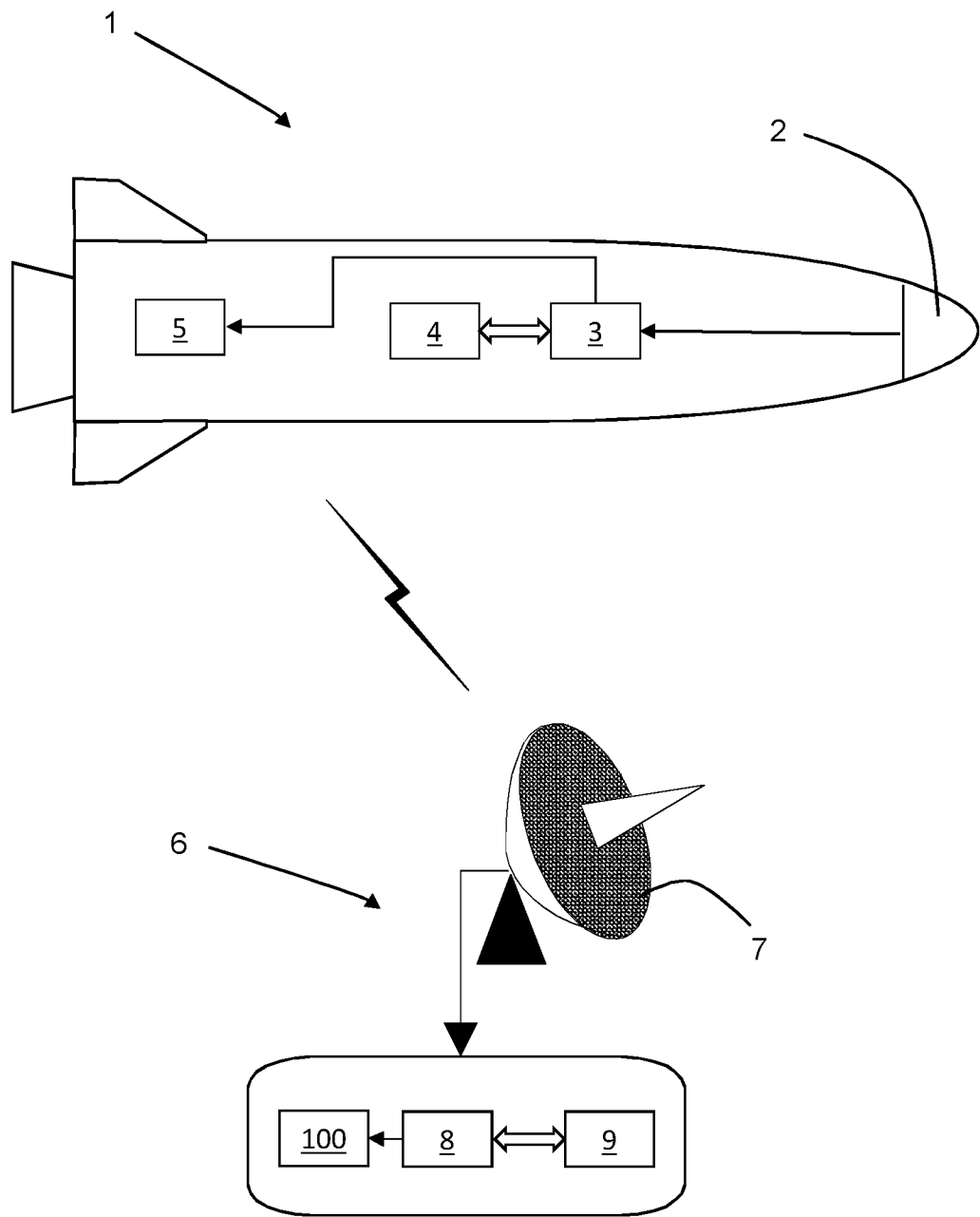


Fig 17

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/GB2023/051588

## A. CLASSIFICATION OF SUBJECT MATTER

INV. H04N19/645 H04N19/60 H04N19/94 H04N19/65  
ADD.

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

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

EPO-Internal, WPI Data

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages        | Relevant to claim No. |
|-----------|-------------------------------------------------------------------------------------------|-----------------------|
| X         | US 2011/074921 A1 (TAKIDUKA HIROSHI [JP]                                                  | 1, 5-9                |
| Y         | ET AL) 31 March 2011 (2011-03-31)<br>paragraph [0226] - paragraph [0228]<br>-----<br>-/-- | 2-4                   |



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents :

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

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

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

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

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

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

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

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

14 July 2023

Date of mailing of the international search report

24/07/2023

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2

NL - 2280 HV Rijswijk

Tel. (+31-70) 340-2040,

Fax: (+31-70) 340-3016

Authorized officer

Schoeyer, Marnix

## INTERNATIONAL SEARCH REPORT

International application No

PCT/GB2023/051588

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                       |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Category*                                            | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Relevant to claim No. |
| Y                                                    | <p>SUGIMOTO K ET AL: "Video coding technology proposal by Mitsubishi Electric",<br/>1. JCT-VC MEETING; 20100415 - 20100423;<br/>DRESDEN; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ),<br/>,<br/>no. JCTVC-A107<br/>12 April 2010 (2010-04-12), XP030232996,<br/>Retrieved from the Internet:<br/>URL:http://phenix.int-evry.fr/jct/doc_end_user/documents/1_Dresden/wg11/JCTVC-A107.zip JCTVC-A107.doc<br/>[retrieved on 2012-03-19]<br/>paragraph [02.2] - paragraph [2.3.2]<br/>paragraph [2.42] - paragraph [2.4.3]<br/>-----</p> | 2-4                   |
| A                                                    | <p>Anonymous: "Triple modular redundancy - Wikipedia",<br/>,<br/>17 September 2020 (2020-09-17),<br/>XP055882878,<br/>Retrieved from the Internet:<br/>URL:https://en.wikipedia.org/w/index.php?title=Triple_modular_redundancy&amp;oldid=978880077<br/>[retrieved on 2022-01-24]<br/>-----</p>                                                                                                                                                                                                                                                                                | 1, 3, 4               |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2023/051588

| Patent document<br>cited in search report | Publication<br>date | Patent family<br>member(s) | Publication<br>date |
|-------------------------------------------|---------------------|----------------------------|---------------------|
| US 2011074921 A1                          | 31-03-2011          | CN 102036044 A             | 27-04-2011          |
|                                           |                     | EP 2306746 A2              | 06-04-2011          |
|                                           |                     | JP 5372687 B2              | 18-12-2013          |
|                                           |                     | JP 2011077785 A            | 14-04-2011          |
|                                           |                     | US 2011074921 A1           | 31-03-2011          |
| -----                                     |                     |                            |                     |