(12) UK Patent (19) GB (11) 2539486 (13) B

(45) Date of B Publication 31.07.2019

(54) Title of the Invention: Encoder, decoder and method employing palette compression

(51) INT CL: ***H04N 19/463*** (2014.01) ***H03M 7/30*** (2006.01) ***H04N 19/13*** (2014.01) ***H04N 19/147*** (2014.01) ***H04N 19/172*** (2014.01) ***H04N 19/186*** (2014.01) ***H04N 19/48*** (2014.01) ***H04N 19/70*** (2014.01) ***H04N 19/91*** (2014.01)

(21) Application No: **1510745.1**

(22) Date of Filing: **18.06.2015**

(43) Date of A Publication **21.12.2016**

(56) Documents Cited:
GB 2371730 A **WO 2015/077720 A1**
US 20140147040 A1
(Q0083) AHG10: A triplet palette mode combining JCTVC-P0108 and JCTVC-P0198, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 27 March 4 April 2014
(Q0063) AHG10: Palette predictor stuffing, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 17th Meeting: Valencia, ES, 27 March 4 April 2014
High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 3, February 2015. JCT-VC T1005.

(58) Field of Search:
As for published application 2539486 A viz:
INT CL **H04N**
Other: **EPODOC; WPI**
updated as appropriate

Additional Fields
INT CL **G06T, H03M, H04N**
Other: **Internet, Patent Fulltext**

(72) Inventor(s):
Ossi Kalevo

(73) Proprietor(s):
Gurulogic Microsystems Oy
Linnankatu 34, Turku FI-20100, Finland

(74) Agent and/or Address for Service:
Basck Ltd
16 Saxon Road, CAMBRIDGE, Cambridgeshire, CB5 8HS, United Kingdom

GB 2539486 B

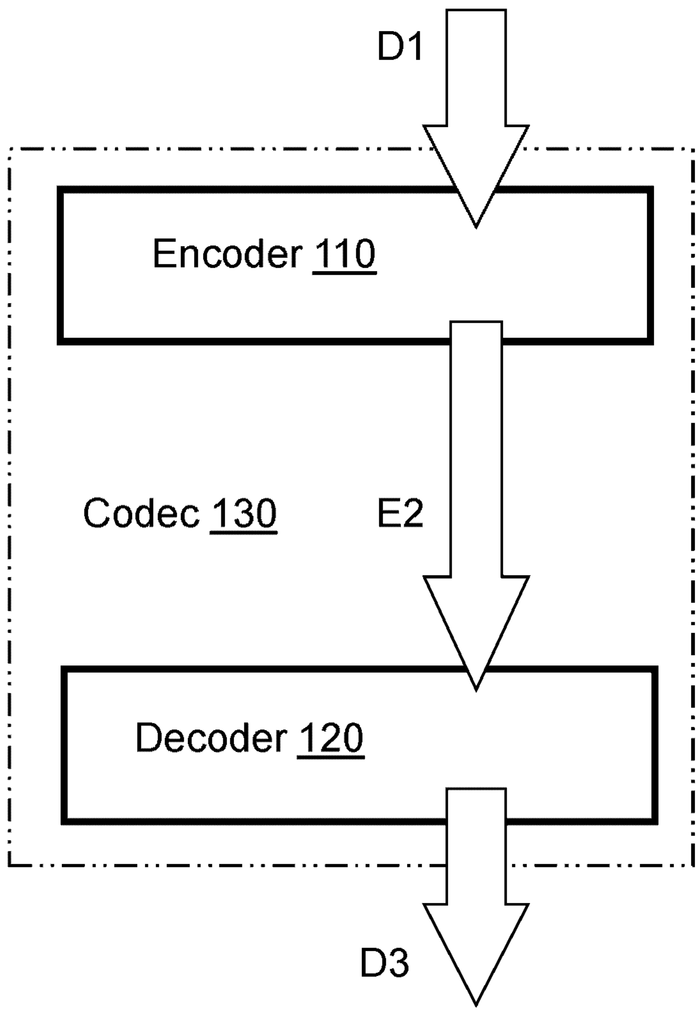


FIG. 1

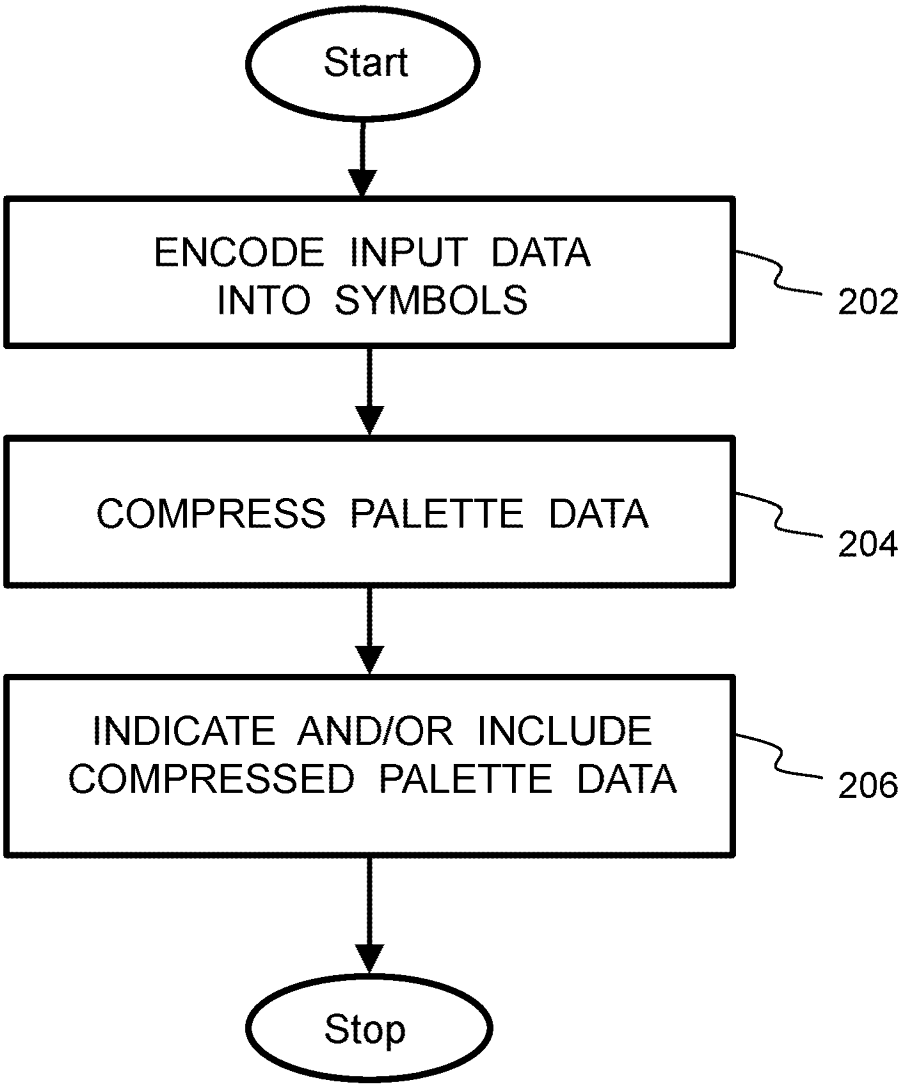

Start
ENCODE INPUT DATA INTO SYMBOLS
202
COMPRESS PALETTE DATA
204
INDICATE AND/OR INCLUDE COMPRESSED PALETTE DATA
206
Stop

FIG. 2

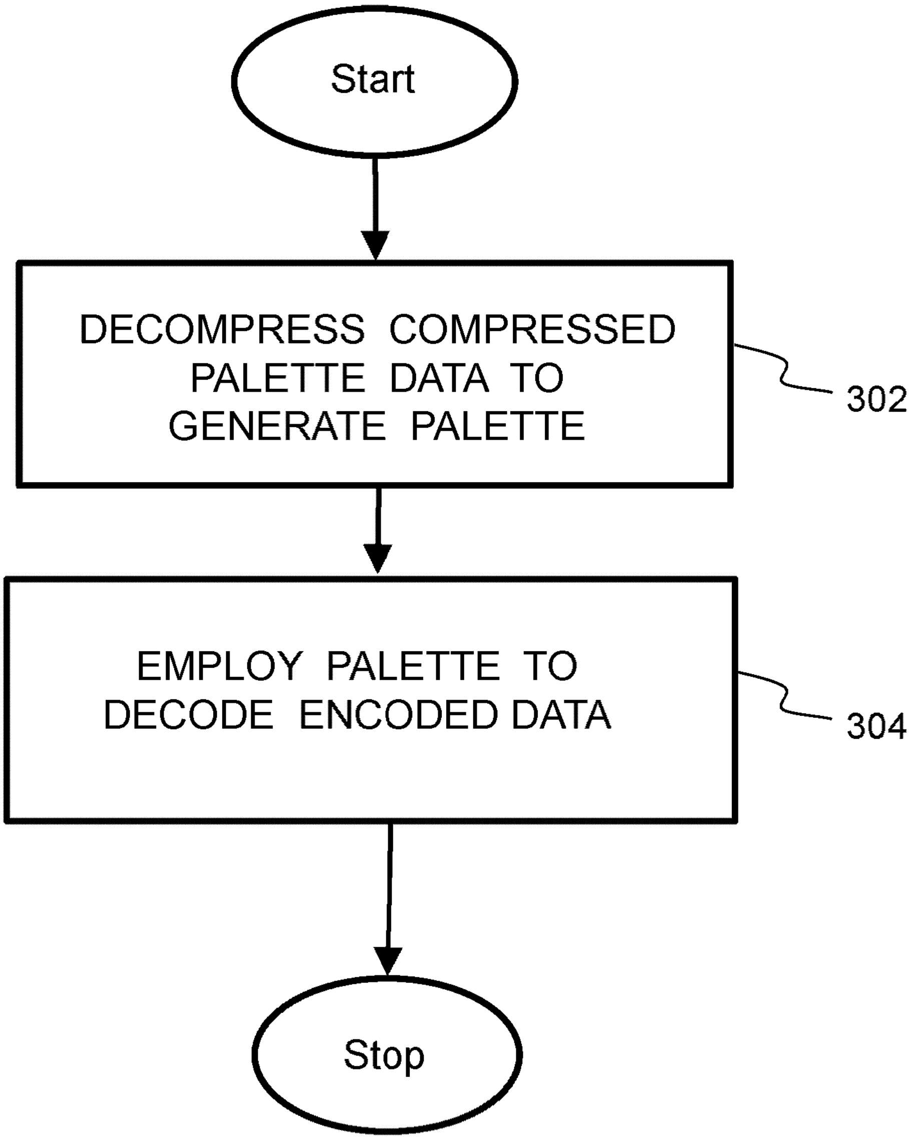

Start
DECOMPRESS COMPRESSED PALETTE DATA TO GENERATE PALETTE
302
EMPLOY PALETTE TO DECODE ENCODED DATA
304
Stop

FIG. 3

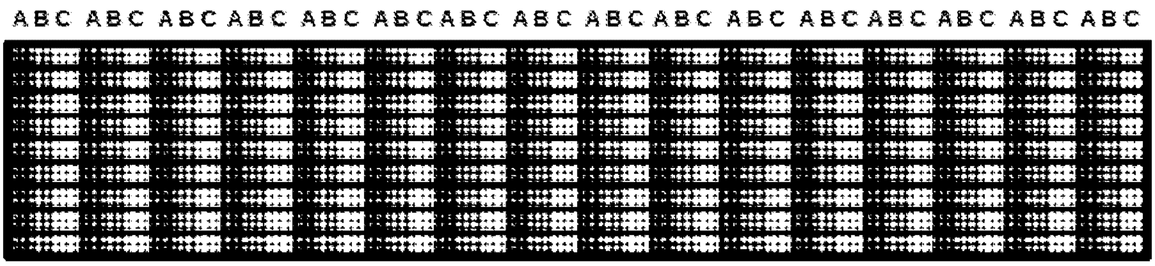

ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC ABC

FIG. 4A

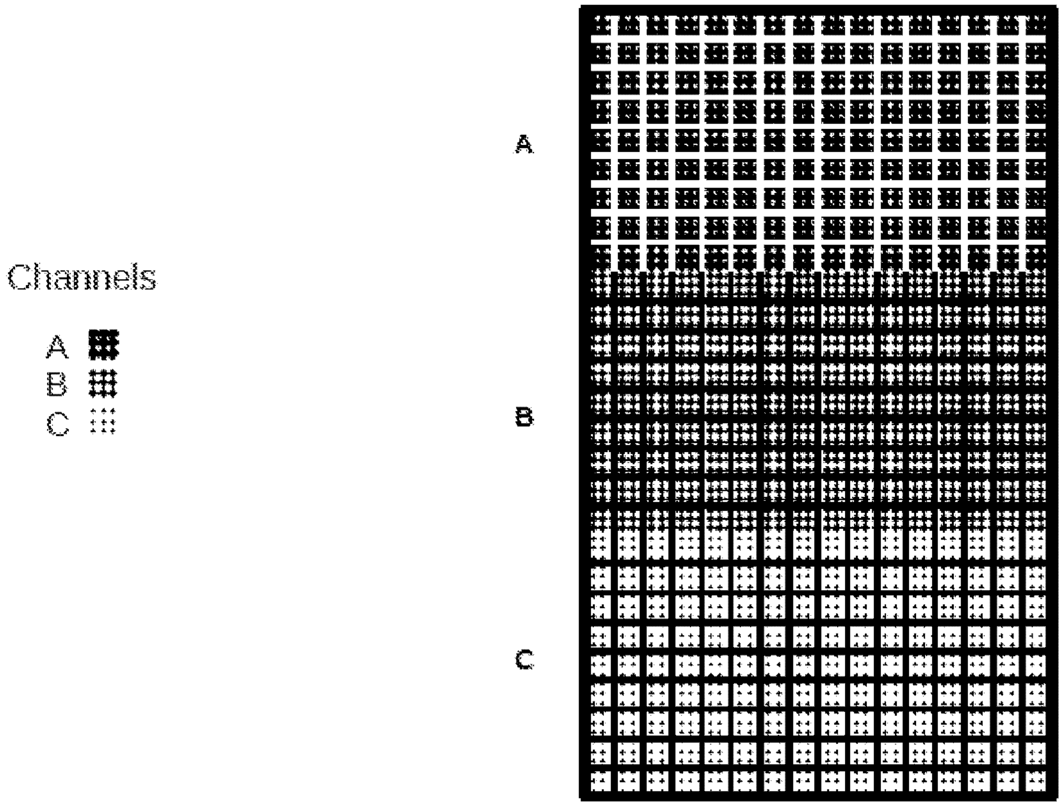

Channels
A
B
C
A
B
C

FIG. 4B

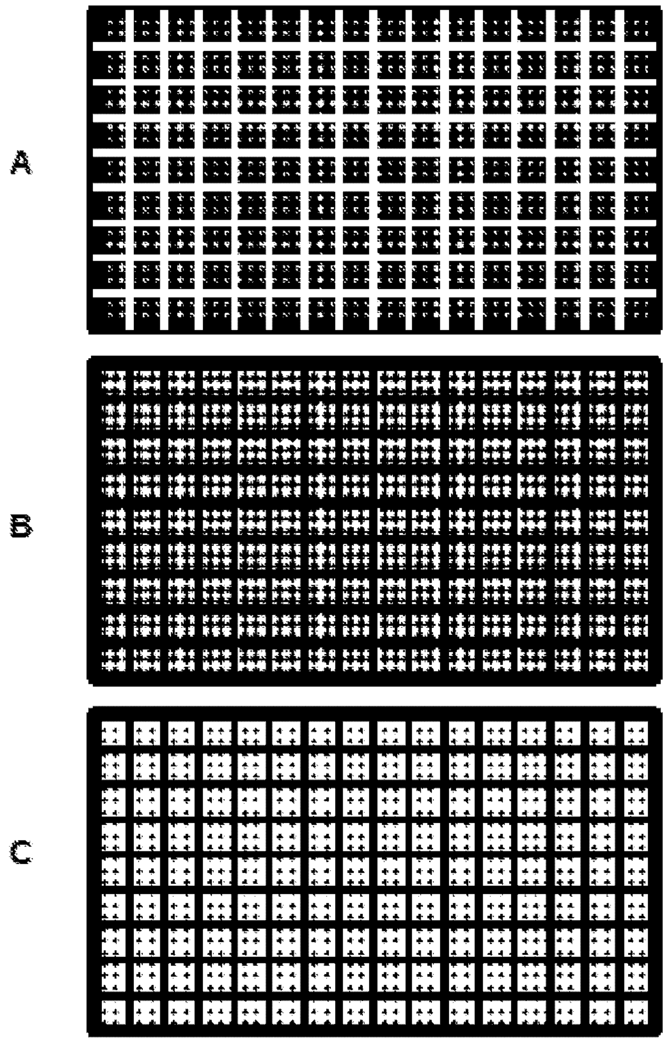
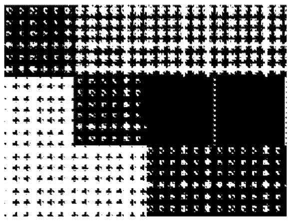


FIG. 4C

Y



Plane

Y

166	102	102	102
97	166	71	71
97	97	166	166

1D-LUT

	0	1	2	3
Y	71	97	102	166

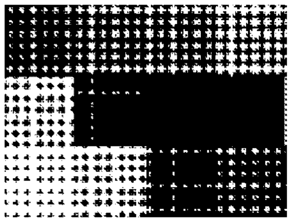
Indexes

Y

3	2	2	2
1	3	0	0
1	1	3	3

FIG. 5

ABC



A	B	C	A	B	C	A	B	C	A	B	C
126	165	60	1	101	205	2	104	203	1	103	205
234	97	62	127	168	60	126	70	205	124	71	203
236	98	61	234	98	63	124	167	60	123	165	62

FIG. 6A

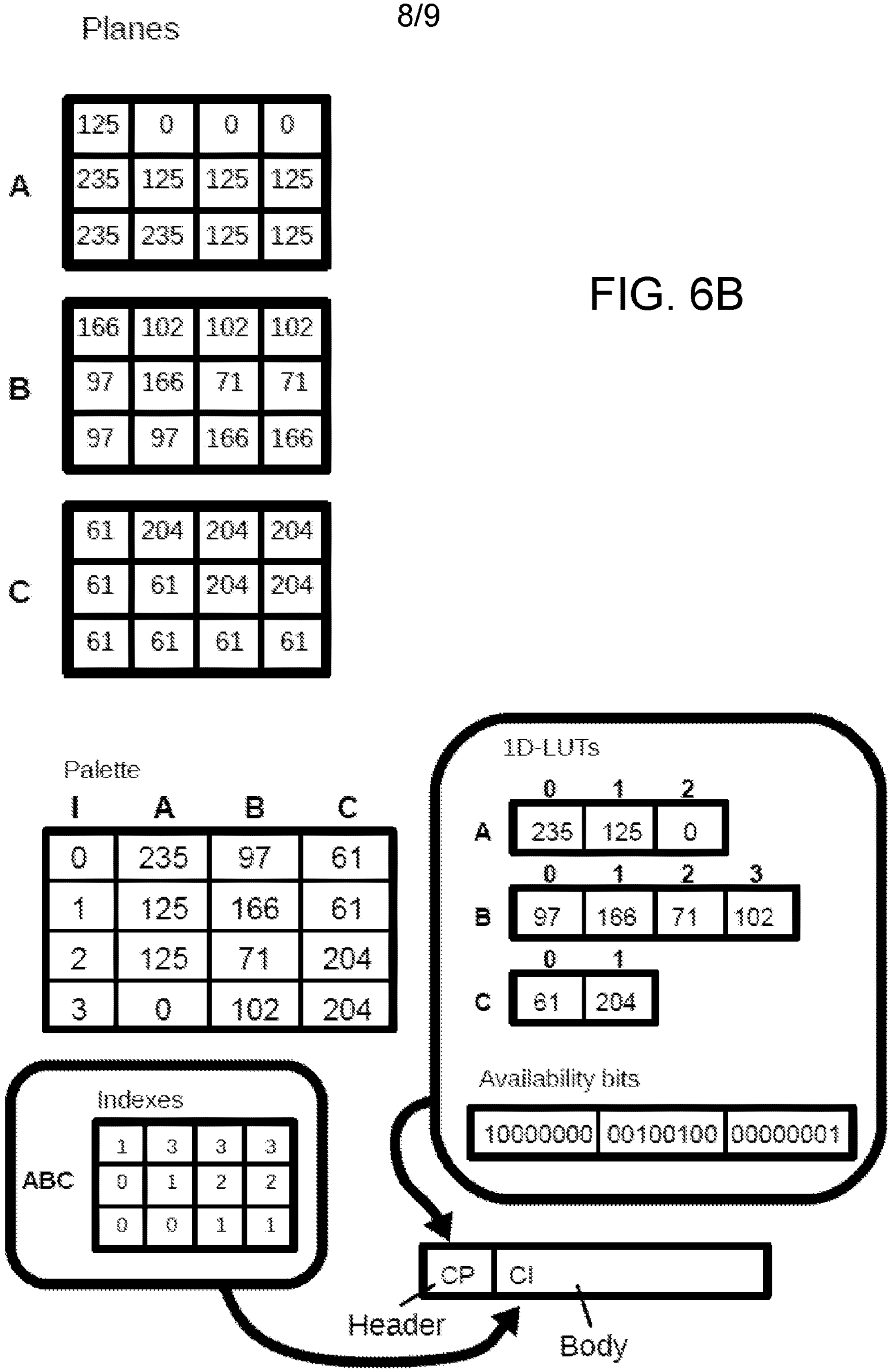
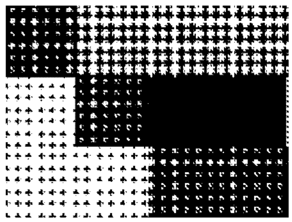

Planes
A
125 0 0 0
235 125 125 125
235 235 125 125
B
166 102 102 102
97 166 71 71
97 97 166 166
C
61 204 204 204
61 61 204 204
61 61 61 61
Palette
I A B C
0 235 97 61
1 125 166 61
2 125 71 204
3 0 102 204
Indexes
ABC
1 3 3 3
0 1 2 2
0 0 1 1
1D-LUTs
0 1 2
A 235 125 0
0 1 2 3
B 97 166 71 102
0 1
C 61 204
Availability bits
10000000 00100100 00000001
CP
CI
Header
Body

FIG. 6B

ABC



A	B	C	A	B	C	A	B	C	A	B	C
125	166	61	0	102	204	0	102	204	0	102	204
235	97	61	125	166	61	125	71	204	125	71	204
235	97	61	235	97	61	125	166	61	125	166	61

FIG. 6C

Intellectual
Property
Office

The following terms are registered trade marks and should be read as such wherever they occur in this document:

App Store (Page 36)

ENCODER, DECODER AND METHOD EMPLOYING PALETTE COMPRESSION

TECHNICAL FIELD

The present disclosure relates to encoders for encoding input data (D1) to generate corresponding encoded data (E2), and corresponding methods of encoding input data (D1) to generate corresponding encoded data (E2). Moreover, the present disclosure relates to decoders for decoding encoded data (E2) to generate corresponding decoded data (D3), and corresponding methods of decoding encoded data (E2) to generate corresponding decoded data (D3). Furthermore, the present disclosure is concerned with computer program products comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute aforesaid methods. Additionally, the present disclosure concerns codecs including at least one aforementioned encoder and at least one aforementioned decoder.

BACKGROUND

Palettes, also known as "*look-up-tables*" (LUT's), are conventionally used to describe some sort of information in some other form. As an example, in a conventional Graphics Interchange Format (GIF) file, a color palette is used to describe, for example, 32/64/128/256 different index values that are used to present original information of the GIF file in a lossy or lossless form. Likewise, conventional Portable Network Graphics format (PNG) files also contain palette information.

Each index value in a GIF file describes, for example, 24-bit Blue-Green-Red (BGR) color values, in a manner that some specific combinations of 8-bit color values are used for describing pixel values of the GIF file in red, green and blue channels. If a given color palette includes 256 index values, 768 bytes (= 3 x 256 bytes) are used in a GIF file to describe the given color palette. It is to be noted here that only a limited number of different 24-bit color values can be described by using a limited number of different index values.

Moreover, it is also conventionally possible to use a palette for other purposes. As an example, a color image containing four channels, for example such as RGBA and CMYK, or only one channel, for example such as in a grayscale image or a separate color channel, can be quantized nonlinearly by using a palette. Such a one-dimensional (1-D) palette, also referred to as a 1-D LUT, makes it possible to present data of the image with fewer bits, namely by employing a smaller data range. Therefore, the data is also susceptible to being compressed more efficiently. Reference [1] presents more detailed background information about color palettes.

It will be appreciated that it is also possible to use a palette for other kinds of data besides images, for example for audio data, video data, measurement data, and so forth. In these cases, a palette (or LUT) employed expresses different types of information other than color values in color pixels. This different type of information includes, for example, an amplitude of an audio signal in a given frequency band; for example, the different type of information describes one or more amplitudes of Fourier harmonic components.

Furthermore, a Color Look-up-Table (CLUT) is often used instead of, or together with, a color palette. A CLUT describes a LUT that enables efficient transformation of used color index values to true colors or vice versa. The CLUT also takes into account a possibility that different true colors are presented with a same color index value, which causes compressed information to be unrecoverable losslessly.

Typically, delivery of a palette uses a large number of bits. As an example, 256 index values for a 32-bit BGRA image require 1 kilobytes (= 4 x 256 bytes) for palette delivery, and additionally a few header bytes for describing, for example, a size of palette data and the number of index values in the palette. Especially when a size of an image is small or the number of index values is large, the number of bits used for palette delivery becomes a significant data overhead.

Prior art implementations used in GIF and PNG file formats do not compress their associated palette at all. The GIF and PNG file formats deliver palette entries of a given palette as interleaved data for each palette entry separately. One reason for adopting such an approach is that, typically, the size of image data is large, and the size of the palette data is relatively small as compared to the size of image data. However, there are still many cases where palette data is a significant piece of data.

Conventionally, it is possible to use well-known palettes that are known in advance to a given encoder and a corresponding given decoder. These well-known palettes do not require palette data to be compressed before delivery from the given encoder to the given decoder. Some examples of such well-known color palettes are described in reference [2].

Furthermore, there are known methods that are available for optimizing an order of palette entries in a palette (for example, see references [3] and [4]). However, these methods do not consider compressing the palette itself.

In a published UK patent document GB2371730A (*Kanji Yoshioka, "Image data storage"*), there is described an image data storage method, wherein If a user wants to store an image acquired, 101, and it is determined, 103, that the data size of the image acquired from the site is larger than a designated storage frame in a memory region, then it is determined, 105, whether or not the number of colors in the palette of the acquired image data can be reduced. If it is possible, the number of palette colors is reduced, 106, to thereby compress the image, and the image data is updated, 108, based on the compressed palette. On the other hand, if it is determined that the number of palette colors cannot be reduced, a control section determines that the image data cannot be stored and abandons, 107, the data, and a display section displays 'storage error'.

In a US patent document US2014147040A1 (*Tatsuya TANAKA et al.*, "*Image Encoding Device, Image Decoding Device, Image Encoding Method, And Image Decoding Method*"), there is described an image encoding device which includes a deciding unit, an assigning unit, and an encoding unit. The deciding unit is configured to determine representative colors for expressing each of pixel blocks into which image data are divided. The assigning unit is configured to assign an index for identifYing the representative color to each pixel in the pixel block. The encoding unit is configured to encode indices and the representative colors, the indices and the representative colors in each pixel box being arranged alternately so that two representative colors are discontinuously encoded.

In a published document "AHG10: A triplet palette mode combining JCTVC-P0108 and JCTVC-P0198" *(Yu-Chen Sun et al.),*it is proposed a contribution which proposes a triplet palette mode combining the two proposals, namely JCTVC-P0108 and JCTVC-P0198. The proposed combination is implemented on top of HM-13.0+RExt-6.0. Experimental results reported show that, compared with HM-13.0+RExt-6.0, X.X%, X.X%, and X.X% BD-rate savings are achieved for SC YUV 444 sequences under AI-Main-Tier, RA-Main-Tier, and LB-Main-Tier, respectively.

In another published document "AhG10: palette predictor stuffing" (*C. Gisquet* et al.), It is proposed in the present contribution to combine the last coded palette and the last predictor palette by stuffing unused elements of the last predictor at the end of the last coded palette, using the resulting palette as predictor for the next palette-coded CU, and adding flags to indicate when there are no predicted elements left. It is reported that the use of such a prediction method provides gains for AI-MT, AI-HT and AI-SHT configurations, compared to AhG10 software of more than 2.3%/3.0%/3.5% for SC classes, and up to 4.3%/5.3%/5.9% on optional classes.

SUMMARY

The present disclosure seeks to provide an improved encoder for encoding input data (D1) to generate corresponding encoded data (E2).

Moreover, the present disclosure seeks to provide an improved encoder for compressing palette information included within input data (D1), when encoding the input data (D1) to generate corresponding encoded data (E2).

Moreover, the present disclosure seeks to provide an improved decoder for decoding encoded data (E2) to generate corresponding decoded data (D3).

Moreover, the present disclosure seeks to provide an improved decoder for decompressing palette information included within encoded data (E2), when decoding the encoded data (E2) to generate corresponding decoded data (D3).

A further aim of the present disclosure is to at least partially overcome at least some of the problems of the prior art, as discussed above.

In a first aspect, embodiments of the present disclosure provide a method of encoding input data (D1) to generate corresponding encoded data (E2), wherein the method includes encoding the input data (D1) into one or more symbols in the encoded data (E2), wherein the one or more symbols represent data as defined by a palette indicated and/or included in the encoded data (E2), characterized in that the method further includes:

compressing data representative of the palette in a lossless manner into the encoded data (E2), by compressing mutually different channels of the palette in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

delivering the compressed data representative of the palette from an encoder to a corresponding decoder, wherein palette entry values of the palette are provided consecutively within the encoded data (E2).

The invention is of advantage in that efficient compression of palette information provides for improved data compression in the encoded data (E2).

When implementing embodiments of the present disclosure, there are potentially employed three mutually different compression methods that, when used alone or in combination, enable compression of palette data. Compression of palette data is very beneficial when a size of input data (D1) is small, or a size of the palette data is large.

The aforementioned method can be used in combination with multiple mutually different encoding methods and standards. As an example, the aforementioned method can be used with a data block encoder described in reference [5]. As another example, the aforementioned method can be used with the Graphics Interchange Format (GIF), the Portable Network Graphics format (PNG) and the like.

Optionally, the method includes generating compressed palette data in a dynamically changing format depending upon content and/or data structure present in the input data (D1), during encoding of the input data (D1). For example, in an event that the input data (D1) changes therethrough, for example when streamed, from a video conference with medical specialists to providing corresponding lists of DNA genetic sequences to support the video conference, providing the palette in a dynamically changing interleaved or planar format is capable of providing more efficient encoding of the input data (D1).

Optionally, the method includes communicating the one or more symbols via a mutually different communication channel to that employed for communicating the compressed palette data.

Optionally, the method includes communicating the one or more symbols via a mutually similar communication channel to that employed for communicating the compressed palette data.

More optionally, the method includes compressing the mutually different channels together or separately in a planar format.

Optionally, the method includes compressing data of a given channel by using one-dimensional look-up table coding.

In a second aspect, embodiments of the present disclosure provide a method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

receiving compressed palette data;

decompressing the compressed palette data indicated and/or included in the encoded data (E2) to generate a palette, palette entry values of the palette being provided consecutively within the encoded data (E2), wherein the compressed palette data is decompressed in a lossless manner, wherein the compressed palette data includes mutually different channels of the palette that are compressed in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

employing the palette to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

Optionally, the method includes generating decompressed palette data in a dynamically changing format during decoding of the encoded data (E2).

More optionally, the method includes employing an inverse of at least one entropy-encoding method employed to decompress the compressed palette data, when the mutually different channels are compressed in an interleaved format.

More optionally, the method includes employing an inverse of an encoding method employed to decompress the mutually different channels together or separately, when the mutually different channels are compressed in a planar format.

More optionally, the method includes generating the palette from the different index values for each of the mutually different channels and the availability information indicative of the combinations of index values used in the palette.

In a third aspect, embodiments of the present disclosure provide an encoder for encoding input data (D1) to generate corresponding encoded data (E2), wherein the encoder is operable to encode the input data (D1) into one or more symbols in the encoded data (E2), wherein the one or more symbols represent data as defined by a palette indicated and/or included in the encoded data (E2),

characterized in that the encoder is operable to:

compress data representative of the palette in a lossless manner into the encoded data (E2), by compressing mutually different channels of the palette in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

deliver the compressed data representative of the palette from the encoder (110) to a corresponding decoder, wherein palette entry values of the palette are provided consecutively within the encoded data (E2).

Optionally, the encoder (110) is operable to generate compressed palette data in a dynamically changing format depending upon content and/or data structure present in the input data (D1), during encoding of the input data (D1).

More optionally, the encoder (110) is operable to communicate the one or more symbols via a mutually different communication channel to that employed for communicating the compressed palette data.

More optionally, the encoder (110) is operable to communicate the one or more symbols via a mutually similar communication channel to that employed for communicating the compressed palette data.

More optionally, the encoder (110) is operable to employ at least one entropy-encoding method to compress the mutually different channels in an interleaved format.

More optionally, the encoder (110) is operable to compress the mutually different channels together or separately in a planar format.

More optionally, the encoder (110) is operable to compress data of a given channel by using one-dimensional look-up table coding.

In a fourth aspect, embodiments of the present disclosure provide a decoder for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

the decoder is operable to receive compressed palette data;

the decoder is operable to decompress the compressed palette data included in the encoded data (E2) to generate a palette, palette entry values of the palette being provided consecutively within the encoded data (E2), wherein the compressed palette data is decompressed in a lossless manner, wherein the compressed palette data includes mutually different channels of the palette that are compressed in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and the decoder (120) is operable to employ the palette to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

More optionally, the decoder (120) is operable to generate decompressed palette data in a dynamically changing format during decoding of the encoded data (E2).

More optionally, the decoder (120) is operable to employ an inverse of at least one entropy-encoding method employed to decompress the compressed palette data, when the mutually different channels are compressed in an interleaved format.

More optionally, the decoder (120) is operable to employ an inverse of an encoding method employed to decompress the mutually different channels together or separately, when the mutually different channels are compressed in a planar format.

More optionally, the decoder (120) is operable to generate the palette from the different index values for the mutually different channels and the availability information indicative of the combinations of index values used in the palette.

In a fifth aspect, embodiments of the present disclosure provide a codec including at least one encoder for encoding input data (D1) to generate corresponding encoded data (E2) pursuant to embodiments of the present disclosure, and at least one decoder for decoding the encoded data (E2) to generate corresponding decoded data (D3) pursuant to embodiments of the present disclosure.

In a sixth aspect, embodiments of the present disclosure provide a computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute any of the aforementioned methods pursuant to embodiments of the present disclosure.

Additional aspects, advantages, features and objects of the present disclosure would be made apparent from the drawings and the detailed description of the illustrative embodiments construed in conjunction with the appended claims that follow.

It will be appreciated that features of the present disclosure are susceptible to being combined in various combinations without departing from the scope of the present disclosure as defined by the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The summary above, as well as the following detailed description of illustrative embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the present disclosure, exemplary constructions of the disclosure are shown in the drawings. However, the present disclosure is not limited to specific methods and apparatus disclosed herein. Moreover, those in the art will understand that the drawings are not to scale. Wherever possible, like elements have been indicated by identical numbers.

Embodiments of the present disclosure will now be described, by way of example only, with reference to the following diagrams wherein:

FIG. 1 is a schematic illustration of an encoder for encoding input data (D1) to generate corresponding encoded data (E2) and a decoder for decoding the encoded data (E2) to generate corresponding decoded data (D3), wherein the encoder and the decoder collectively form a codec, in accordance with an embodiment of the present disclosure;

FIG. 2 is a schematic illustration of a flow chart depicting steps of a method of encoding input data (D1) to generate corresponding encoded data (E2), in accordance with an embodiment of the present disclosure;

FIG. 3 is a schematic illustration of a flow chart depicting steps of a method of decoding encoded data (E2) to generate corresponding decoded data (D3), in accordance with an embodiment of the present disclosure;

FIGs. 4A, 4B and 4C is a group of illustrations of a data or palette including mutually different channels of the data or palette that are to be compressed in various ways, wherein

FIG. 4A is an illustration of an interleaved format, wherein the channels are to be compressed together, in accordance with an embodiment of the present disclosure;

FIG. 4B is an illustration of a planar format, wherein the channels are to be compressed together, in accordance with an embodiment of the present disclosure;

FIG. 4C is an illustration of a planar format, wherein the channels are to be compressed separately, in accordance with an embodiment of the present disclosure;

FIG. 5 is an illustration of data that is modified with 1-dimensional (1-D) look-up-table (LUT) for compression, in accordance with an embodiment of the present disclosure; and

FIGs. 6A, 6B and 6C collectively are an illustration of data, compressed palette based upon the data, the palette, 1-D LUT for each channels and availability bits of different available 1-D LUT value combinations, wherein some of the 1-D LUT value combinations are used in the palette, in accordance with an embodiment of the present disclosure.

In the accompanying diagrams, an underlined number is employed to represent an item over which the underlined number is positioned or an item to which the underlined number is adjacent. A non-underlined number relates to an item identified by a line linking the non-underlined number to the item.

DETAILED DESCRIPTION OF EMBODIMENTS

The following detailed description illustrates embodiments of the present disclosure and ways in which they can be implemented. Although some modes of carrying out the present disclosure have been disclosed, those skilled in the art would recognize that other embodiments for carrying out or practising the present disclosure are also possible.

In a first aspect, embodiments of the present disclosure provide a method of encoding input data (D1) to generate corresponding encoded data (E2), wherein the method includes encoding the input data (D1) into one or more symbols in the encoded data (E2), wherein the one or more symbols represent data as defined by a palette indicated and/or included in the encoded data (E2), characterized in that the method further includes:

compressing data representative of the palette in a lossless manner into the encoded data (E2), by compressing mutually different channels of the palette in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

delivering the compressed data representative of the palette from an encoder to a corresponding decoder, wherein palette entry values of the palette are provided consecutively within the encoded data (E2).

Throughout the present disclosure, the one channel or mutually different channels of the palette are hereinafter interchangeably referred to as "*palette data*".

It will be appreciated that a palette can be used for mutually different types of data, for example, such as image data, video data, audio data, genomic data, measurement data, seismic data, magnetic resonance imaging (MRI) data, deoxyribonucleic acid (DNA) data, ribonucleic acid (RNA) data, biometric data, and so forth. As an example, an audio palette can be used to express an amplitude of an audio signal in a given frequency band. However, it will be appreciated that, contemporarily, color palettes are commonly used for images and represent an example application of embodiments of the present disclosure.

The aforementioned method can be implemented via a given encoder. Optionally, the compressed palette data is delivered from the given encoder to a corresponding given decoder by using one or more data files, or by streaming data from the given encoder to the given decoder. Optionally, the one or more data files can be conveyed via a data carrier, for example, such as an optical disc data memory and/or a solid state data memory.

According to an embodiment of the present disclosure, the method includes communicating the one or more symbols via a mutually different communication channel to that employed for communicating the compressed palette data. In an example, the compressed palette data is provided as a Universal Serial Bus (USB) dongle for attaching to a personal computer in a form of a physical "*access key*" for enabling decoding of data to be executed in the personal computer, whereas the one or more symbols are streamed via a data communication network to the personal computer. Similar considerations pertain to a wireless communication device, wherein the compressed palette data is downloaded in a software application, wherein the software application is used in the wireless communication device to decode one or more symbols included in data that is streamed wirelessly to the wireless communication device. Optionally, the software application handles periodic updating of the compressed palette data.

According to another embodiment, the method includes communicating the one or more symbols via a mutually similar communication channel to that employed for communicating the compressed palette data. Such a scenario pertains, for example, when streaming data to a wireless communication device, for example a smart phone, or to a monitoring arrangement of a seismic geological survey system, or a down-borehole video monitoring system for use in petrochemicals industry.

Moreover, the aforementioned method can be used in combination with multiple mutually different encoding methods and standards. As an example, the aforementioned method can be used with a data block encoder described in reference [5]. As another example, the aforementioned method can be used with the Graphics Interchange Format (GIF), the Portable Network Graphics format (PNG) and the like.

According to an embodiment of the present disclosure, when compressing in the interleaved format, the method includes employing at least one entropy-encoding method to compress the one channel or mutually different channels in the interleaved format. In the interleaved format, index values of all the mutually different channels of the palette are compressed together. Optionally, in this regard, the at least one entropy-encoding method is used to compress index values of the mutually different channels in a sequence in which these index values occur in the interleaved format. As an example, the mutually different channels can be compressed using Huffman coding, Variable-Length Coding (VLC), range coding, or via a reference to a database; an example of such compression of data via a reference to a database is described in a patent application GB2509055A, see reference [8], filed by Gurulogic Microsystems Oy.

It will be appreciated that when only one channel is interleaved, then in practice only that channel is interleaved which in practice is equal to the planar format.

According to an embodiment of the present disclosure, when compressing in the planar format, the method includes compressing the mutually different channels together or separately. As index values of one channel are typically independent of index values of other channels, the mutually different channels can be compressed as planar channels using optionally different encoding methods. Such a planar format potentially results in a relatively simpler data structure in the encoded data (E2), but also potentially requires less computing effort to implement. In contradistinction, the aforementioned interleaved format is potentially more robust to unauthorized eavesdropping of the encoded data (E2), for example in situations where data security is important. Optionally, encoders and decoders pursuant to the present disclosure are operable to switch dynamically between employing an interleaved format and a planar format when communicating data, for example palette-related data, therethrough.

For a given channel, a suitable encoding method can be selected based upon an inspection of data or index values of the given channel. As an example, if the given channel includes large index values that occur repeatedly, variable length coding (VLC) can be used to compress the data or index values of the given channel. As another example, if the given channel includes small index values that occur sequentially, Delta coding or ODelta coding can be used to compress the data or index values of the given channel. Herein, the term "*Delta coding*" refers to a method of storing or transmitting data in a form of differences between sequential data rather than complete data files or data values, while the term "*ODelta coding*" refers to a differential form of encoding based upon wraparound in a binary or integer counting regime, for example as described in the United Kingdom patent document GB 1412937.3.

Moreover, optionally, when compressing in the planar format, the method includes compressing data of a given channel by using one-dimensional look-up-table (1-D LUT) coding. In the 1-D LUT coding, original index values of the given channel are replaced by new index values specified in a 1-D LUT. The 1-D LUT coding is particularly beneficial in a case where the original index values are numbers that are sparsely distributed in a whole dynamic range of the index values, or are not numbers that are replaced by indexing to numbers.

Beneficially, the new index values are relatively small as compared to the original index values. Therefore, optionally, Delta or ODelta coding is used to compress these new index values.

Moreover, optionally, a total number of different index values is reduced when the 1-D LUT coding is used with linear or nonlinear quantization. In this regard, a count of original index values is taken into account when quantized index values are defined for a 1-D LUT, so as to reduce, for example minimize, distortion due to the quantization. When lossy compression is used, it is beneficial to take into account how much distortion will be caused to the original index values when the original index values are replaced with quantized index values and reconstructed. As an example, a squared Euclidean distance can be used to represent the distortion caused by lossy compression of palette data, and the compression can then be efficiently optimized by using a Rate Distortion (RD) method, wherein an RD value is calculated as a distortion caused by compression added by "*lambda*" times a rate required for the palette data and indexed data, wherein "*lambda*" is a Lagrange multiplier.

A quantized index value for the 1-D LUT can be one of the original index values, or can be any other value that lies between a range of the original index values, for example an average, a weighted average, a median, or a mode.

According to an embodiment of the present disclosure, the method includes compressing the mutually different channels by delivering different index values, for example as a 1-D LUT, for each channel (hereinafter referred to as "*channel data*") separately, and delivering availability information indicative of combinations of index values that are used in the palette. This is particularly beneficial when the palette is large, but each channel contains only a few different index values, namely originally or after quantization or using a 1-D LUT.

It will be appreciated that, in a case of 1-D LUT, all the aforementioned palette compression methods are in fact similar, wherein 1-D LUT values are one after another.

Optionally, the availability information is delivered using availability bits. Delivering the availability information is particularly beneficial when all possible combinations of the index values are not available in the palette.

It will be appreciated that an order of combinations is also defined, so as to enable a given encoder and a corresponding given decoder to create a similar palette based upon the channel data and the availability bits. Optionally, the order of combinations is pre-defined and fixed. Alternatively, optionally, information indicative of the order of combinations is delivered from the given encoder to the corresponding given decoder.

Moreover, optionally, the order of combinations is changed, so as to improve further the compression of the palette data. It is to be noted here that the order of combinations has a large influence on the compression of the input data (D1). Therefore, when the order of combinations is changed, it is beneficial to take into account both a total number of bits required to deliver the encoded data (E2) and a total number of bits required to deliver the compressed palette data.

Moreover, optionally, if amounts of the used and unused combinations are very different, the availability bits can be compressed by employing, for example, range coding with, namely as symbols, or without, namely as bits or symbols, an Entropy Modifying (EM) encoding method that is described in reference [6].

Moreover, it will be appreciated that when different channels of a given palette contain different numbers of mutually different index values, the number of different index values for each channel is delivered separately. Furthermore, the number of availability bits depends on a multiplication of the number of different index values in each channel. Optionally, in this regard, 1-D LUT coding with quantization is particularly beneficial to use. The 1-D LUT coding can be used with quantization to reduce the number of different index values to a smaller number. As a result, the number of possible combinations for the channels of the given palette decreases considerably, which, in turn, reduces the number of availability bits required for describing combinations actually used in the given palette.

In a second aspect, embodiments of the present disclosure provide a method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

receiving compressed palette data;

decompressing compressed palette data included in the encoded data (E2) to generate a palette, palette entry values of the palette being provided consecutively within the encoded data (E2), wherein the compressed palette data is decompressed in a lossless manner, wherein the compressed palette data includes mutually different channels of the palette that are compressed in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

employing the palette to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

The aforementioned method can be implemented via a given decoder. The decoder is optionally implemented using digital hardware, or by using a computing device that is suitably configured, for example by executing decoder software products. Optionally, the computing device employs a high-speed reduced instruction set computing (RISC) device, which is well suited for performing at a high speed, in a repetitive manner, relatively simpler types of data processing.

Optionally, the compressed palette data is received via one or more data files, or via streaming from a given encoder to the given decoder. Optionally, the compressed data is provided by using a combination of a physical data carrier and data streaming via a data communication network.

Moreover, the aforementioned method can be used in combination with multiple mutually different decoding methods and standards. As an example, the aforementioned method can be used with a data block decoder described in reference [7]. As another example, the aforementioned method can be used with the Graphics Interchange Format (GIF), the Portable Network Graphics format (PNG) and the like.

According to an embodiment of the present disclosure, when the palette data has been compressed in the interleaved format, the method includes employing an inverse of at least one entropy-encoding method that was employed at a given encoder to decompress the compressed palette data. As an example, the compressed palette data can be decompressed using Huffman decoding, variable-length decoding, range decoding, or via a reference to a database; example of data compression performed via a reference to a database is described in a patent application GB2509055A, filed by Gurulogic Microsystems Oy (see reference [8]).

08 05 19

According to an embodiment of the present disclosure, when the palette data has been compressed in the planar format, the method includes employing an inverse of an encoding method that was employed at the given encoder to decompress the mutually different channels together or separately, for example each of the mutually different channels are decompressed separately.

According to an embodiment of the present disclosure, the method includes generating the palette data from the different index values, for example as a 1-D LUT, for each channel and the availability information indicative of the combinations of index values that are used in the palette. Optionally, in this regard, the method includes receiving information indicative of an order of the combinations from the given encoder.

Next, embodiments of the present disclosure will be described in greater detail.

For illustration purposes only, there will now be provided examples of how a given palette can be compressed pursuant to embodiments of the present disclosure.

In an example, input data (D1) is an interleaved 24-bit RGB image that has symbols as follows:

(i) **R(1), G(1), B(1), R(2), G(2), B(2), ... R(N), G(N), B(N)**

or

(ii) **R(1)G(1)B(1), R(2)G(2)B(2), ... R(N)G(N)B(N)**

Hereinabove, the symbols are 8-bit symbols in (i) and 24-bit symbols in (ii);
'R(N)' denotes an index value of a red color component of an N^{th} pixel;
'G(N)' denotes an index value of a green color component of an N^{th} pixel; and
'B(N)' denotes an index value of a blue color component of an N^{th} pixel.

Optionally, color palettes can be created for each channel separately, namely separately for R, G and B channels of the RGB image. Alternatively, optionally, a single color palette can be created for all channels together.

As an example, for a color palette in the GIF or PNG format, palette data can include 256 RGB palette elements corresponding to 256 24-bit RGB colors, wherein each palette element, namely each palette entry, is expressed using 24 bits. Without compression, delivery of such a color palette requires 6144 bits (= 256 x 24 bits = 768 bytes) to be delivered in addition to a palette header.

In operation, palette data is compressed pursuant to embodiments of the present disclosure, thereby enabling an efficient delivery of the palette data as compared to conventional methods, where palette data is delivered in an uncompressed manner.

There will now be described below three compression methods that, when used alone or in combination, at least partially overcome at least some of the problems of the prior art, as discussed earlier, namely:

(1) a first compression method, wherein palette data is entropy-encoded in an interleaved format (namely together);

(2) a second compression method, wherein palette data is encoded in a planar format (namely together or separately); and

(3) a third compression method, wherein different index values, for example as a 1-dimensional look-up-table (1-D LUT), for each channel of a palette are indicated, together with availability information indicative of combinations of index values used in the palette.

Firstly, in respect of embodiments of the present disclosure, entropy-encoding of palette data in an interleaved format will next be described.

Optionally, in this regard, at least one entropy-encoding method is employed to compress the palette data in the interleaved format. As an example, Huffman coding, VLC, range coding or a database (for example, see reference [8] concerning database coding that can be employed) can be used to compress index values of the palette data as they occur in the interleaved format.

For illustration purposes only, there will now be considered an example of a two-channel palette that includes 11 symbols having 22 index values as follows:

(2, 10), (4, 20), (6, 10), (8, 20), (10, 10), (10, 20), (11, 20), (12, 10), (13, 20), (14, 10), (20, 20)

Without compression, when each of these 22 index values are represented by five bits, 110 bits are required to be delivered in addition to palette header for palette delivery from a given encoder to a corresponding given decoder. Alternatively, without compression, when each of these 22 index values are represented by eight bits, 176 bits (as are used in, for example, GIF or PNG) are required to be delivered in addition to palette header for palette delivery from a given encoder to a corresponding given decoder.

In the above example, a count of each of index values ‘10’ and ‘20’ is seven, which is much higher than a count of any of remaining index values, namely index values ‘2’, ‘4’, ‘6’, ‘8’, ‘11’, ‘12’, ‘13’ and ‘14’. In other words, the index values ‘10’ and ‘20’ are more probable than the remaining index values. Therefore, Variable-Length Coding (VLC) or range coding can be used to compress the example palette for delivery with a smaller number of bits.

As an example, using VLC, occurrences of the remaining index values can be represented by a VLC bit code ‘0’, while occurrences of the more-probable index values ‘10’ and ‘20’ can be represented by VLC bit codes ‘10’ and ‘11’, respectively. Moreover, each of the remaining index values can be represented using four bits. Thus, a total number of bits required for the palette delivery is only 68 bits, as calculated below:

8 x (1+4) + (7+7) x 2 bits = 68 bits

Moreover, a VLC table is also required to be delivered from the given encoder to the given decoder, but it requires fewer than 42 bits. As a result, at least some compression benefits are achieved when the example palette is encoded using VLC.

Secondly, in respect of embodiments of the present disclosure, encoding of palette data in a planar format will next be described.

In this regard, each channel of a given palette is compressed separately into a planar format.

Continuing from the previous example of the two-channel palette, a first channel of the palette includes 11 index values in a sequence as follows:

2, 4, 6, 8, 10, 10, 11, 12, 13, 14, 20

, while a second channel of the palette includes 11 index values in a sequence as follows:

10, 20, 10, 20, 10, 20, 20, 10, 20, 10, 20

The sequence of data of the first channel can be compressed, for example, using Delta coding or range coding. As an example, the first channel can be compressed using Delta coding with a first prediction value of ‘0’ (zero) to yield Delta values as follows:

2, 2, 2, 2, 2, 0, 1, 1, 1, 1, 6

The sequence of data of the second channel can be compressed, for example, using VLC. As an example, using VLC, occurrences of the index values ‘10’ and ‘20’ can be represented by VLC bit codes ‘0’ and ‘1’, respectively.

The Delta values can be delivered using three bits per Delta value, while the VLC bit codes can be delivered using only one bit per VLC bit code. Thus, apart from a VLC table that is required to be delivered from the given encoder to the given decoder, a total number of bits required for the palette delivery in the planar format is only 44 bits, as calculated below:

11 x 3 + 11 x 1 bits = 44 bits

Now, the total number of bits required in the planar format is even less than the total number of bits required in the aforementioned interleaved format. Sometimes, the index values in different channels are similar and they can be compressed together in a similar manner in which the entropy coding is employed in the aforementioned interleaved format example. However, an only difference is that the order of index values is now different, and this potentially provides considerable benefits, for example when using Delta coding together with range coding.

Moreover, optionally, data of a given channel can be compressed by using 1-D LUT coding, for example, as will be described in detail below.

When using a 1-D LUT, a total number of different index values can be reduced if lossy compression of the data is employed. In other words, the total number of different index values can be reduced when the 1-D LUT is used with quantization. In this regard, a count of original index values is taken into account when quantized index values are defined for the 1-D LUT, so as to minimize distortion due to the quantization.

For illustration purposes only, there will now be considered an example of how the 1-D LUT coding can be used with quantization pursuant to embodiments of the present disclosure. In the example, a given channel of a given palette includes 12 index values in a sequence as follows:

178, 2, 229, 230, 75, 2, 2, 2, 178, 77, 230, 230

Hereinabove, the given channel includes only six different original index values, namely the original index values ‘2’, ‘75’, ‘77’, ‘178’, ‘229’ and ‘230’. As an example, four quantized index values can be defined for the six original index values as follows:

a quantized index value ‘2’ for the original index value ‘2’,
a quantized index value ‘76’ for the original index values ‘75’ and ‘77’,
a quantized index value ‘178’ for the original index value ‘178’, and
a quantized index value ‘230’ for the original index values ‘229’ and ‘230’.

Moreover, a lossy 1-D LUT can be used to map these quantized index values to new index values, for example, as follows:

Quantized Index Value	New Index Value
2	0
76	1

178	2
230	3

It is to be noted here that the new index values in the 1-D LUT are not required to be expressed in a specific order. However, often it is beneficial to have the new index values to be expressed in a specific order.

Each of the new index values can be delivered using only two bits, while each of the quantized index values can be delivered using eight bits. Therefore, the lossy 1-D LUT can be delivered from the encoder to the corresponding decoder with only 40 bits, as calculated below:

4 x 2 + 4 x 8 bits = 40 bits

Delivery of the 1-D LUT can be optimized even more when the new index values are expressed in a specific order. When it is predefined that the new index values are in a specific order, they do not need to be delivered from the encoder to the decoder; in such a case, only the quantized index values are delivered using delta coding that uses a “0” value as a first prediction value. Those Delta values are then calculated as below:

2, 74, 102, 52

Each of these Delta values can be represented using 7 bits. In other words, only 4 x 7 bits = 28 bits are needed to deliver the described 1-D LUT from the encoder to the decoder.

In the illustrated example, when the lossy 1-D LUT is used for encoding the given channel, the new index values are provided in a sequence as follows:

2, 0, 3, 3, 1, 0, 0, 0, 2, 1, 3, 3

When the original index values of the given channel are delivered without compression, 96 bits (= 12 x 8 bits) are required to be delivered. However, when the new index values are delivered, only 24 bits (= 12 x 2 bits) are required to be delivered in addition to 28 bits for delivering the 1-D LUT.

Furthermore, optionally, Delta or ODelta coding is beneficially used to compress these new index values, as the new index values are relatively small as compared to the original index values. Such Delta or ODelta coding of the new index values potentially reduces a total number of bits required to be delivered for delivering the 1-D LUT, the given palette and the data of the given channel.

Thirdly, in respect of embodiments of the present disclosure, encoding of palette data in a format that indicates different index values, for example as a 1-D LUT, for each channel of a palette, together with availability information, will next be described.

Optionally, in this regard, palette data is compressed by delivering different index values, for example as a 1-D LUT, for each channel (hereinafter referred to as "*channel data*") separately, and delivering availability bits denoting which combinations of index values are used in the palette. This is particularly beneficial when the palette is large, but each channel contains only a few different index values, namely originally or after quantization or using a 1-D LUT.

It will be appreciated that an order of combinations is also defined, so as to enable a given encoder and a corresponding given decoder to create a similar palette based upon the channel data and the availability bits.

As an example, if there are three channels in a given palette and eight different index values that are used for each channel, then following are required to be delivered:

(i) eight bits for representing the number of different index values, namely eight in this example,

(ii) 64 bits (= 8 x 8 bits) for representing the eight different index values, for example as a 1-D LUT, for each channel separately, and

(iii) 512 availability bits (= 8 x 8 x 8 bits) for representing combinations that are used in the given palette.

If, in this example, there are 256 used and 256 unused combinations, a total number of bits required to deliver these 256 different 24-bit palette elements is 712 bits, as calculated below:

8 + 3 x 64 + 512 = 712 bits = 89 bytes

Thus, the total number of bits required to be delivered for the palette delivery is much less as compared to conventional methods, such as the aforementioned GIF or PNG, namely only 712 bits (89 bytes) are required to be delivered, instead of 6144 bits (768 bytes).

For illustration purposes only, there will now be considered an example of how a given palette can be delivered using availability bits pursuant to embodiments of the present disclosure. In the example, the given palette corresponds to three channels as follows:

(i) a first channel having three index values ‘0’, ‘1’ and ‘2’,

(ii) a second channel having four index values ‘0’, ‘1’, ‘2’ and ‘3’, and

(iii) a third channel having two index values ‘0’ and ‘1’.

As a result, there are 24 (= 3 x 4 x 2) possible combinations of index values for the given palette that can be represented in a following order:

(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (0, 2, 0), (0, 2, 1), (0, 3, 0), (0, 3, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1), (1, 2, 0), (1, 2, 1), (1, 3, 0), (1, 3, 1), (2, 0, 0), (2, 0, 1), (2, 1, 0), (2, 1, 1), (2, 2, 0), (2, 2, 1), (2, 3, 0), (2, 3, 1)

As an example, the availability bits are used to represent used and unused combinations as follows:

11110111 11101111 11011110

From the availability bits, it is determined at a given decoder that there are 20 used combinations and 4 unused combinations. In other words, the combinations (0, 2, 0), (1, 1, 1), (2, 1, 0), and (2, 3, 1) are not used in the given palette. The remaining combinations, namely the used combinations, can then be indicated with new index values, for example from '0' to '19' in the same order.

In this example, the total number of bits required to be delivered for the palette delivery is only 120 bits, as calculated below:

$$3 \times 8 + (3 + 4 + 2) \times 8 + 24 \text{ bits} = 120 \text{ bits}$$

, wherein eight bits are required for delivering the number of different index values for each of the three channels, 72 bits are required for delivering the different index values, and 24 bits are needed for availability bits.

Thus, only 120 bits are required to be delivered, instead of delivering 480 bits (= 20 x 24 bits) for the 20 used combinations of the given palette.

Moreover, optionally, the total number of bits can be further reduced by at least one of:

(i) employing a suitable encoding method based upon an inspection of real dynamics, namely a data range of the index values, for example such as Delta coding, and/or

(ii) using range coding, with or without entropy modification (EM), for the availability bits.

It is evident that the three compression methods described in the foregoing, when used alone or in combination, improve the palette delivery from a given encoder to a corresponding given decoder. These compression methods, either alone or in combination, are beneficially used, for example, when a size of input data (D1) is small, or a size of the palette data is large.

Pursuant to embodiments of the present disclosure, at least two of the three compression methods are optionally used together, so as to further improve the palette delivery from the given encoder to the corresponding given decoder. As an example, channel data generated by the third compression method can be delivered using Delta coding.

Moreover, it will be appreciated that, in the third compression method, when different channels of a given palette contain different numbers of mutually different index values, the number of different index values, for example as a 1-D LUT, for each channel is delivered together or separately. Moreover, the number of availability bits depends on a multiplication of the number of different index values in each channel. Optionally, in this regard, 1-D LUT coding with quantization is particularly beneficial to use with the third compression method. The 1-D LUT coding can be used with quantization to reduce the number of different index values to a smaller number. As a result, the number of possible combinations for the channels of the given palette decreases considerably, which, in turn, reduces the number of availability bits required for describing combinations actually used in the given palette.

In a third aspect, embodiments of the present disclosure provide an encoder for encoding input data (D1) to generate corresponding encoded data (E2), wherein the encoder is operable to encode the input data (D1) into one or more symbols in the encoded data (E2), wherein the one or more symbols represent data as defined by a palette indicated and/or included in the encoded data (E2),

characterized in that the encoder is operable to:

compress data representative of the palette in a lossless manner into the encoded data (E2), by compressing mutually different channels of the palette in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

deliver the compressed data representative of the palette from the encoder to a corresponding decoder, wherein palette entry values of the palette are provided consecutively within the encoded data (E2).

Optionally, the encoder is operable to deliver the compressed palette data to a corresponding given decoder by using one or more data files, or by streaming data to the given decoder. In an example, a combination of physical data memory and data streaming is employed when delivering the encoded data (E2). The physical data memory is provided, for example, as a USB dongle for a personal computer, namely in a form of a physical "*access key*" to enable the personal computer to decode the encoded data (E2).

According to an embodiment of the present disclosure, the encoder is operable to communicate the one or more symbols via a mutually different communication channel to that employed for communicating the compressed palette data.

According to another embodiment of the present disclosure, the encoder is operable to communicate the one or more symbols via a mutually similar communication channel to that employed for communicating the compressed palette data.

Moreover, the aforementioned encoder can be used together with multiple mutually different encoders. As an example, the aforementioned encoder can be used with a data block encoder described in reference [5]. As another example, the aforementioned encoder can be used with the Graphics Interchange Format (GIF), the Portable Network Graphics format (PNG) and the like.

According to an embodiment of the present disclosure, when compressing in the interleaved format, the encoder is operable to employ at least one entropy-encoding method to compress one channel or mutually different channels in the interleaved format. Optionally, in this regard, the encoder is operable to use the at least one entropy-encoding method to compress index values of the mutually different channels in a sequence in which these index values occur in the interleaved format. As an example, the encoder is operable to compress the mutually different channels using Huffman coding, Variable-Length Coding (VLC), range coding, or via a reference to a database, for example as described in reference [8].

According to an embodiment of the present disclosure, when compressing in the planar format, the encoder is operable to compress the mutually different channels together or separately, as described earlier.

Moreover, optionally, when compressing in the planar format, the encoder is operable to compress data of a given channel by using 1-D LUT coding, as described earlier.

According to an embodiment of the present disclosure, the encoder is operable to compress the palette data by delivering different index values, for example as a 1-D LUT, for each channel separately, and delivering availability information indicative of combinations of index values that are used in the palette. This is particularly beneficial when the palette is large, but each channel contains only a few different index values, namely originally or after quantization or using a 1-D LUT.

Optionally, the availability information is delivered using availability bits. Delivering the availability information is particularly beneficial when all possible combinations of the index values are not available in the palette.

It will be appreciated that an order of combinations is also defined, so as to enable the encoder and a corresponding decoder to create a similar palette based upon the channel data and the availability bits. Optionally, the order of combinations is pre-defined and fixed. Alternatively, optionally, information indicative of the order of combinations is delivered from the encoder to the corresponding decoder.

In a fourth aspect, embodiments of the present disclosure provide a decoder for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

the decoder is operable to receive compressed palette data;

the decoder is operable to decompress compressed palette data included in the encoded data (E2) to generate a palette, palette entry values of the palette being provided consecutively within the encoded data (E2), wherein the compressed palette data is decompressed in a lossless manner, wherein the compressed palette data includes mutually different channels of the palette that are compressed in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

the decoder is operable to employ the palette to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

Optionally, the decoder is operable to receive the compressed palette data via one or more data files, or via streaming from a given encoder.

Moreover, the aforementioned decoder can be used together with multiple mutually different decoders. As an example, the aforementioned decoder can be used with a data block decoder described in reference [7]. As another example, the aforementioned decoder can be used with the Graphics Interchange Format (GIF), the Portable Network Graphics format (PNG) and the like.

According to an embodiment of the present disclosure, when the palette data has been compressed in the interleaved format, the decoder is operable to employ an inverse of at least one entropy-encoding method that was employed at a given encoder to decompress the compressed palette data. As an example, the decoder is operable to decompress the compressed palette data using Huffman decoding, variable-length decoding, range decoding, or via a reference to a database, for example as described in reference [8].

According to an embodiment of the present disclosure, when the palette data has been compressed in the planar format, the decoder is operable to employ an inverse of an encoding method that was employed at the given encoder to decompress one channel or the mutually different channels together or separately.

According to an embodiment of the present disclosure, the decoder is operable to generate the palette data from the different index values, for example as a 1-D LUT, for each channel and the availability information indicative of the combinations of index values that are used in the palette. Optionally, in this regard, the decoder is operable to receive information indicative of an order of the combinations from the given encoder.

In a fifth aspect, embodiments of the present disclosure provide a codec including at least one encoder for encoding input data (D1) to generate corresponding encoded data (E2) pursuant to embodiments of the present disclosure, and at least one decoder for decoding the encoded data (E2) to generate corresponding decoded data (D3) pursuant to embodiments of the present disclosure. Such a codec is, for example, employed in scientific instruments for sensing a given region, for generating the input data (D1), for encoding the input data (D1) to generate corresponding encoded data (E2), for example for storage in data memory of the scientific instruments, and for providing review of the encoded data (E2), for example replay of the encoded data (E2), spatially locally at the scientific instruments. The scientific instruments include, for example medical ultrasonic sensing apparatus, MRI imagers, endoscopic inspection devices, dental X-ray (Röntgen) apparatus, and so forth, but not limited thereto.

An example codec has been provided in conjunction with FIG. 1 as explained in more detail below. The codec includes at least one encoder and at least one decoder.

The at least one encoder employs a method of encoding input data (D1) as described in the foregoing including compression of palette data. The at least one encoder is thus operable to encode the input data (D1) to generate corresponding encoded data (E2).

Moreover, the at least one decoder is operable to perform an inverse of operations executed in the at least one encoder, to decode the encoded data (E2) to generate corresponding decoded data (D3).

Optionally, the decoded data (D3) is exactly similar to the input data (D1), as in a lossless mode of operation. Alternatively, optionally, the decoded data (D3) is substantially similar to the input data (D1), as in a lossy mode of operation. Yet alternatively, optionally, the decoded data (D3) is different to the input data (D1), for example by way of transcoding or via use of one or more transformations, for example, such as a color conversion, a format conversion, an upscaling conversion, a downscaling conversion, a cropping conversion, a rotation conversion, a flipping conversion but not limited thereto, but retains substantially similar information present in the input data (D1); for example, the decoded data (D3) is usefully made different to the input data (D1) when reformatting of the decoded data (D3) is also required, for example to be compatible with different types of communication platforms, software layers, communication devices, display devices and so forth.

A color conversion can be deemed to be optimal for following reasons:

(i) it is able to achieve as good a reconstruction as is possible; and/or

(ii) it is able to execute the optimal color conversion as regards, for example, properties of a display device to be used for presenting the decoded data (D3).

However, an essential issue is the following: if a palette exists that contains values, for example in the RGB color space, and it is desired to reconstruct an image in the YUV color space, then it is considerably more efficient firstly to transform the palette values from the RGB color space to the YUV color space and then to map those YUV values from the transformed palette, than it would be if the values were firstly mapped from the original RGB palette and then a thereby generated image were transformed into the YUV color space. It will be appreciated that the amount of necessary operations is considerably smaller in the first option here.

The at least one encoder includes a data processing arrangement for processing the input data (D1) to generate the corresponding encoded data (E2) pursuant to embodiments of the present disclosure. Optionally, the data processing arrangement of the at least one encoder is implemented by employing at least one Reduced Instruction Set Computing (RISC) processor that is operable to execute program instructions as elucidated earlier. RISC processors are operable, for example, to implement relatively simple data manipulations at a very great speed, whilst simultaneously utilizing relatively little electrical power, namely a characteristic that is highly advantageous when embodiments of the present disclosure are implemented in mobile battery-powered apparatus, for example "*black box*" flight recorders for aircraft.

Furthermore, optionally, the at least one encoder is operable to communicate the encoded data (E2) to a data server and/or data storage for storing in a database. The data server and/or data storage is arranged to be accessible to the at least one decoder, which is beneficially compatible with the at least one encoder, for subsequently decoding the encoded data (E2).

In some examples, the at least one decoder is optionally operable to access the encoded data (E2) from the data server and/or data storage.

In alternative examples, the at least one encoder is optionally operable to stream the encoded data (E2) to the at least one decoder, either via a data carrier or a data communication network or via a direct connection. Moreover, it is to be noted that a device equipped with a hardware-based or software-based encoder can also communicate directly with another device equipped with a hardware-based or software-based decoder.

In yet other alternative examples, the at least one decoder is optionally implemented so as to retrieve the encoded data (E2) from a non-transitory (namely non-transient) computer-readable storage medium, such as a hard drive and a Solid-State Drive (SSD).

The at least one decoder includes a data processing arrangement for processing the encoded data (E2) to generate the corresponding decoded data (D3) pursuant to embodiments of the present disclosure. Optionally, the data processing arrangement of the at least one decoder is implemented by employing at least one RISC processor that is operable to execute program instructions as elucidated earlier; such a RISC processor is capable of performing relatively simpler concatenated operations at a very high speed, and is suitable for decoding data provided in a streamed format, for example in real-time. RISC processors are contemporarily employed in smart phones for performing data processing of heterodyned wireless signals at high speed, and enable embodiments of the present disclosure to be conveniently implemented on such smart phones, for example.

When embodiments of the present disclosure are implemented in a multicasting manner, there is a plurality of such decoders that are employed.

Optionally, the codec is implemented within a single device. Alternatively, optionally, the codec is effectively implemented between multiple devices. Optionally, the codec is implemented as custom-designed digital hardware, for example via use of one or more Application-Specific Integrated Circuits (ASIC's). Alternatively or additionally, optionally, the codec is implemented using computing hardware that is operable to execute program instructions, for example provided to the computing hardware on a non-transient (non-transitory) machine-readable data carrier.

As an example, the at least one encoder and/or the at least one decoder can be beneficially employed in consumer electronics apparatus, wireless communication apparatus and associated systems, cameras, smart phones, tablet computers, personal computers, scientific measuring apparatus, flight recorders, communications equipments, display devices, videoconferencing equipments, satellites, but not limited thereto.

In a sixth aspect, embodiments of the present disclosure provide a computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute any of the aforementioned methods pursuant to embodiments of the present disclosure.

Optionally, the computer-readable instructions are downloadable from a software application store, for example, from an "*App store*[RTM]" to the computerized device, for example to a smart phone.

Next, embodiments of the present disclosure will be described with reference to figures.

Referring to FIG. 1, embodiments of the present disclosure concern:

(i) an encoder **110** for encoding input data (D1) to generate corresponding encoded data (E2), and corresponding methods of encoding the input data (D1) to generate the encoded data (E2);

(ii) a decoder **120** for decoding the encoded data (E2) to generate corresponding decoded data (D3), and corresponding methods of decoding the encoded data (E2) to generate the decoded data (D3); and

(iii) a codec **130** including a combination of at least one encoder and at least one decoder, for example a combination of the encoder **110** and the decoder **120**.

FIG. 1 is merely an example, which does not unduly limit the scope of the claims herein. It is to be understood that the specific designation for the codec **130** is provided as an example and is not to be construed as limiting the codec **130** to specific numbers, types, or arrangements of encoders and decoders. A person skilled in the art will recognize many variations, alternatives, and modifications of embodiments of the present disclosure.

Referring now to FIG. 2, there is provided a flow chart depicting steps of a method of encoding input data (D1) to generate corresponding encoded data (E2), in accordance with an embodiment of the present disclosure. The method is depicted as a collection of steps in a logical flow diagram, which represents a sequence of steps that can be implemented in hardware, software, or a combination thereof, for example as aforementioned.

At a step **202**, the input data (D1) is encoded into one or more symbols in the encoded data (E2). The one or more symbols represent data as defined by a palette.

At a step **204**, one channel or mutually different channels of the palette, namely palette data, are compressed:

(i) in an interleaved format (namely together);

(ii) in a planar format (namely together or separately); and/or

(iii) in a format that indicates different index values, for example as a 1-dimensional look-up-table (1-D LUT), for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette.

Any combination of (i) to (iii) are potentially employed, for example in a dynamically changing manner as given input data (D1) is being encoded to generate the encoded data (E2), *mutatis mutandis* in a dynamic manner in a corresponding decoder.

At a step **206**, the compressed palette data is indicated and/or included in the encoded data (E2).

The steps **202** to **206** are only illustrative and other alternatives can also be provided where one or more steps are added, one or more steps are removed, or one or more steps are provided in a different sequence without departing from the scope of the claims herein.

Referring now to FIG. 3, there is provided a flow chart depicting steps of a method of decoding encoded data (E2) to generate corresponding decoded data (D3), in accordance with an embodiment of the present disclosure. The method is depicted as a collection of steps in a logical flow diagram, which represents a sequence of steps that can be implemented in hardware, software, or a combination thereof, for example as aforementioned.

At a step **302**, compressed palette data indicated and/or included in the encoded data (E2) is decompressed to generate a palette. The compressed palette data includes one channel or mutually different channels of the palette that are compressed:

(i) in an interleaved format (namely together);

(ii) in a planar format (namely together or separately); and/or

(iii) in a format that indicates different index values, for example as a 1-dimensional look-up-table (1-D LUT), for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette.

At a step **304**, the palette is employed to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

The steps **302** to **304** are only illustrative and other alternatives can also be provided where one or more steps are added, one or more steps are removed, or one or more steps are provided in a different sequence without departing from the scope of the claims herein.

Referring next to FIG. 4A, there is shown an illustration of example data or palette expressed via use of channels A, B and C in an interleaved format, wherein the data or the palette associated with the channels A, B and C are to be compressed mutually together, in accordance with an embodiment of the present disclosure. Moreover, in FIG. 4B, there is shown an illustration of example data or palette expressed via use of channels A, B and C in a planar format, wherein the data or the palette associated with the channels A, B, and C are to be compressed mutually together, in accordance with an embodiment of the present disclosure. Furthermore, in FIG. 4C, there is shown an illustration of example data or palette, wherein the data or the palette associated with the channels A, B and C are in a planar format and are to be compressed mutually separately, in accordance with an embodiment of the present disclosure. The example data represents, for example, image data, audio data, sensor data, genetic data, but not limited thereto.

Referring next to FIG. 5, data denoted by "Y" is, for example, color-defining data associated with one or more images. When expressed in a planar format, the data "Y" has integer values, as shown, in a range of 71 to 166. The integer values are expressible in a 1-dimensional look-up-table (1-D LUT), having values 71, 97, 102 and 166, which are referenced by index values 0, 1, 2 and 3, respectively. Using the index values, the color-defining values can also be defined efficiently in a planar format, by using fewer bits, with reference to the 1-D LUT. There is thereby provided an efficient method of encoding the data denoted by “Y”, in accordance with an embodiment of the present disclosure.

In FIG. 6A, an example of representing data to be encoded in an interleaved format for channels A, B and C is shown at a bottom region of the diagram and an image is shown at a top region of the diagram.

An example of a lossy method of encoding data, which is illustrated in FIG. 6A, is provided in FIG. 6B, in accordance with an embodiment of the present disclosure. Referring next to FIG. 6B, the data to be encoded is represented in three planes associated with channels A, B and C. For the values present in the data to be encoded, it is feasible to generate a lossy palette defined by an index parameter value I, and values for the channels A, B and C, as illustrated. The palette can, for example, be described by a collection of 1-dimensional look-up-tables (1-D LUT), wherein each of the channels A, B and C has its associated 1-D LUT as shown. By employing the palette and its index parameter values I, it is feasible to represent the data to be encoded as lossy encoded data, wherein the encoded data includes a header portion that includes compressed palette, denoted by “*CP*”, and a body portion that includes compressed data as compressed index values based upon palette, denoted by “*CI*”. Different portions of CP and/or CI data are optionally further encoded by using entropy encoding methods, for example, such as range coding or VLC coding.

An example of a decoded data generated from the lossy encoded data of FIG. 6B is illustrated in FIG. 6C, wherein the decoded data is represented, for channels A, B and C, in an interleaved format shown at a bottom region of the diagram and as an image shown at a top region of the diagram.

Methods associated with FIG. 4A to FIG. 6B are usefully employed in an encoder for encoding input data (D1) to generate corresponding encoded data (E2), and an inverse of the methods is usefully employed when decoding the encoded data

(E2) to generate corresponding decoded data (D3). An example of decoded data is shown in FIG. 6C. In FIG. 5, two upper images represent the data to be encoded and corresponding decoded data as per a lossless mode of operation.

FIGs. 4A to 6C are merely examples, which do not unduly limit the scope of the claims herein. A person skilled in the art will recognize many variations, alternatives, and modifications of embodiments of the present disclosure.

Modifications to embodiments of the invention described in the foregoing are possible without departing from the scope of the invention as defined by the accompanying claims. Expressions such as “including”, “comprising”, “incorporating”, “consisting of”, “have”, “is” used to describe and claim the present invention are intended to be construed in a non-exclusive manner, namely allowing for items, components or elements not explicitly described also to be present. Reference to the singular is also to be construed to relate to the plural. Numerals included within parentheses in the accompanying claims are intended to assist understanding of the claims and should not be construed in any way to limit subject matter claimed by these claims.

REFERENCES

[1] Palette (computing) - Wikipedia, the free encyclopedia
(accessed April 29, 2015)
URL: http://en.wikipedia.org/wiki/Palette_%28computing%29

[2] List of color palettes - Wikipedia, the free encyclopedia
(accessed April 29, 2015)
URL: http://en.wikipedia.org/wiki/List_of_color_palettes

[3] A Method for Increasing Compression Ratio of Palette Color Images
(accessed April 29, 2015)
URL: http://www.penguin.cz/~fojtik/palette/palette.htm

[4] IEEE Xplore Abstract - A survey on palette reordering methods for improving the compression of color-indexed images
(accessed April 29, 2015)
URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1344032&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F83%2F29595%2F01344032)

[5] Encoder and Method
(accessed April 29, 2015);
URL: http://gurulogic.com/files/US8675731B2.pdf

[6] Entropy Modifier and Method
(accessed April 29, 2015)
URL: http://gurulogic.com/files/US8754791B1.pdf

[7] Decoder and Method
(accessed April 29, 2015)
URL: http://gurulogic.com/files/US2014044191A1.pdf

[8] Database coding method, as described in GB patent application GB1222240.2, published as GB2509055A.

CLAIMS

We claim:

1. A method of encoding input data (D1) to generate corresponding encoded data (E2), wherein the method includes encoding the input data (D1) into one or more symbols in the encoded data (E2), wherein the one or more symbols represent data as defined by a palette indicated and/or included in the encoded data (E2), characterized in that the method further includes:

compressing data representative of the palette in a lossless manner into the encoded data (E2), by compressing mutually different channels of the palette in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

delivering the compressed data representative of the palette from an encoder to a corresponding decoder, wherein palette entry values of the palette are provided consecutively within the encoded data (E2).

2. The method of claim 1, characterized in that the method includes generating compressed palette data in a dynamically changing format depending upon content and/or data structure present in the input data (D1), during encoding of the input data (D1).

3. The method of claim 1 or 2, characterized in that the method includes communicating the one or more symbols via a mutually different communication channel to that employed for communicating the compressed palette data.

4. The method of claim 1 or 2, characterized in that the method includes communicating the one or more symbols via a mutually similar communication channel to that employed for communicating the compressed palette data.

5. The method of any of claims 1 to 4, characterized in that the method includes compressing data of a given channel by using one-dimensional look-up table coding.

6. A method of decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that the method includes:

receiving compressed palette data;

decompressing the compressed palette data included in the encoded data (E2) to generate a palette, palette entry values of the palette being provided consecutively within the encoded data (E2), wherein the compressed palette data is decompressed in a lossless manner, wherein the compressed palette data includes mutually different channels of the palette that are compressed in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

employing the palette to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

7. The method of claim 6, characterized in that the method includes generating decompressed palette data in a dynamically changing format during decoding of the encoded data (E2).

8. The method of any of claims 6 or 7, characterized in that the method includes generating the palette from the different index values for each of the mutually different channels and the availability information indicative of the combinations of index values used in the palette.

9. An encoder (110) for encoding input data (D1) to generate corresponding encoded data (E2), wherein the encoder (110) is operable to encode the input

data (D1) into one or more symbols in the encoded data (E2), wherein the one or more symbols represent data as defined by a palette indicated and/or included in the encoded data (E2),

characterized in that the encoder is operable to:

compress data representative of the palette in a lossless manner into the encoded data (E2), by compressing mutually different channels of the palette in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

deliver the compressed data representative of the palette from the encoder (110) to a corresponding decoder, wherein palette entry values of the palette are provided consecutively within the encoded data (E2).

10. The encoder (110) of claim 9, characterized in that the encoder (110) is operable to generate compressed palette data in a dynamically changing format depending upon content and/or data structure present in the input data (D1), during encoding of the input data (D1).

11. The encoder (110) of claim 9 or 10, characterized in that the encoder (110) is operable to communicate the one or more symbols via a mutually different communication channel to that employed for communicating the compressed palette data.

12. The encoder (110) of claim 9 or 10, characterized in that the encoder (110) is operable to communicate the one or more symbols via a mutually similar communication channel to that employed for communicating the compressed palette data.

13. The encoder (110) of any of claims 9 to 12, characterized in that the encoder (110) is operable to compress data of a given channel by using one-dimensional look-up table coding.

14. A decoder (120) for decoding encoded data (E2) to generate corresponding decoded data (D3), characterized in that:

the decoder is operable to receive compressed palette data;

the decoder (120) is operable to decompress the compressed palette data included in the encoded data (E2) to generate a palette, palette entry values of the palette being provided consecutively within the encoded data (E2), wherein the compressed palette data is decompressed in a lossless manner, wherein the compressed palette data includes mutually different channels of the palette that are compressed in a format that indicates different index values for each of the mutually different channels, together with availability information indicative of combinations of index values used in the palette; and

the decoder (120) is operable to employ the palette to decode one or more symbols in the encoded data (E2) to generate the decoded data (D3).

15. The decoder (120) of claim 14, characterized in that the decoder (120) is operable to generate decompressed palette data in a dynamically changing format during decoding of the encoded data (E2).

16. The decoder (120) of claim 14 or 15, characterized in that the decoder (120) is operable to generate the palette from the different index values for the mutually different channels and the availability information indicative of the combinations of index values used in the palette.

17. A codec (130) including at least one encoder (110) of any of claims 9 to 13 for encoding input data (D1) to generate corresponding encoded data (E2),

and at least one decoder (120) of any of claims 14 to 16 for decoding the encoded data (E2) to generate corresponding decoded data (D3).

18. A computer program product comprising a non-transitory computer-readable storage medium having computer-readable instructions stored thereon, the computer-readable instructions being executable by a computerized device comprising processing hardware to execute a method as claimed in any of claims 1 to 8.