

IMAGE PROCESSING DEVICE AND METHOD FOR OPERATING IMAGE PROCESSING DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

5 **[0001]** This application claims the benefit of priority to Korean Patent Application No. 10-2018-0010083, filed on January 26, 2018, and Korean Patent Application No. 10-2018-0041781, filed on April 10, 2018, in the Korean Intellectual Property Office, the disclosures of which are incorporated by reference in their entireties herein.

TECHNICAL FIELD

10 **[0002]** The present disclosure relates to an image processing device and method for operating an image processing device.

BACKGROUND

15 **[0003]** More and more applications demand high-definition video images and high-frame rate images. Accordingly, the amount of data accessed from a memory (i.e., the bandwidth) storing these images by various multimedia Intellectual Property (IP) blocks of image processing devices has greatly increased.

20 **[0004]** Each image processing device has limited processing capability. When the bandwidth increases, the processing capability of the image processing device may reach this limit. Accordingly, a user of the image processing device may experience a decrease in speed while recording or playing a video image.

SUMMARY

25 **[0005]** At least one embodiment of the present inventive concept provides an image processing device that executes optimized image data compression.

[0006] At least one embodiment of the present inventive concept provides a method for operating an image processing device that executes optimized image data compression.

30 **[0007]** According to an exemplary embodiment of the present inventive concept, there is provided an image processing device including a multimedia intellectual property (IP) block

which processes image data; a memory; and a frame buffer compressor (FBC) which compresses the image data to generate compressed data and stores the compressed data in the memory. The frame buffer compressor includes a logic circuit which determines a combination of a quantization parameter (QP) table and an entropy table and controls compression of the image data based on the determined combination of the QP table and the entropy table.

[0008] According to an exemplary embodiment of the present inventive concept, there is provided an image processing device including a multimedia IP block which processes image data; a memory; and a frame buffer compressor (FBC) which compresses the image data to generate compressed data and stores the compressed data in the memory. The frame buffer compressor includes a logic circuit which determines a quantization parameter (QP) table including a maximum of 16 entries and an entropy table determined by a maximum of 4 k values for entropy encoding, and controls compression of the image data based on the determined combination of the QP table and the entropy table.

[0009] According to an exemplary embodiment of the present inventive concept, there is provided a method for operating an image processing device including converting image data into predicted image data including prediction data and residual data; determining a quantization parameter (QP) table including a predetermined number of entries; quantizing the predicted image data, using the determined QP table; determining an entropy table, using a predetermined number of k values for entropy encoding; and performing the entropy coding on the quantized image data, using the determined entropy table to generate compressed data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings, in which:

[0011] FIGS. 1 to 3 are block diagrams for explaining an image processing device according to some embodiments of the present inventive concept;

[0012] FIG. 4 is a block diagram for explaining a frame buffer compressor of FIGS. 1 to 3 in detail;

[0013] FIG. 5 is a block diagram for explaining an encoder of FIG. 4 in detail;

[0014] FIG. 6 is a block diagram for explaining a decoder of FIG. 4 in detail;

[0015] FIGS. 7 to 10 are schematic diagrams for explaining an operation of the image processing device according to exemplary embodiments of the present inventive concept;

[0016] FIGS. 11 to 14 are schematic diagrams for explaining an operation of the image processing device according to exemplary embodiments of the present inventive concept;

5 [0017] FIG. 15 is a diagram for explaining advantageous effects of the image processing device and the method for operating the image processing device according to exemplary embodiments of the present inventive concept; and

[0018] FIG. 16 is a flowchart illustrating a method for operating the image processing device according to exemplary embodiments of the present inventive concept.

10

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0019] FIGS. 1 to 3 are block diagrams for explaining an image processing device according to exemplary embodiments of the present inventive concept.

15 [0020] Referring to FIG. 1, the image processing device according to exemplary embodiments of the present inventive concept includes a multimedia IP (Intellectual Property) 100 (e.g., an IP block, and IP core, a circuit etc.), a frame buffer compressor (FBC) 200 (e.g., a circuit, a digital signal processor, etc.), a memory 30, and a system bus 400.

[0021] In an embodiment, the multimedia IP 100 is a part of the image processing device that directly executes the image processing of the image processing device. The multimedia IP
20 100 may include a plurality of modules for recording and reproducing images such as camcoding and playback of video images.

[0022] The multimedia IP 100 receives first data (e.g., image data) from an outside source such as a camera, and converts the first data into second data. For example, the first data may be moving image data or raw image data. The second data is data generated by the multimedia
25 IP 100, and may include data resulting from the multimedia IP 100 processing the first data. The multimedia IP 100 may repeatedly store the second data in the memory 300 and update the second data via various steps. The second data may include all the data used in these steps. The second data may be stored in the memory 300 in the form of third data. Therefore, the second data may be data before being stored in the memory 300 or after read from the memory
30 300. This will be explained in more detail below.

[0023] In an exemplary embodiment, the multimedia IP 100 includes an image signal processor ISP 110, a shake correction module G2D 120, a multi-format codec MFC 130, a GPU 140 and a display 150. However, the present inventive concept is not limited thereto. That is, the multimedia IP 100 may include at least one of the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150. The multimedia IP 100 may be implemented by a processing module (e.g., a processor) that accesses the memory 300 in order to process data representing moving or static images.

[0024] The image signal processor 110 receives the first data, and pre-processes the first data to convert the first data into the second data. In an exemplary embodiment, the first data is an RGB type image source data. For example, the image signal processor 110 may convert the first data of the RGB type into a YUV type second data.

[0025] In an embodiment, the RGB type data means a data format which expresses colors on the basis of the three primary colors of light. That is, it is a type that expresses images, using three kinds of colors of red (RED), green (GREEN), and blue (BLUE). In contrast, the YUV type means a data format which separately expresses brightness, that is, a luma signal and a chroma signal. That is, Y means the luma signal, and U(Cb) and V(Cr) mean chroma signals, respectively. U means a difference between the luma signal and the blue signal component, and V means a difference between the luma signal and the red signal component.

[0026] The YUV type data may be acquired by converting the RGB type data using a conversion formula. For example, a conversion formula such as $Y = 0.3R + 0.59G + 0.11B$, $U = (B - Y) \times 0.493$, $V = (R - Y) \times 0.877$ may be used to convert the RGB type data into the YUV type data.

[0027] Since human eyes are sensitive to the luma signal, but are less sensitive to the color signal, the YUV type data may be more easily compressed than RGB type data. As a result, the image signal processor 110 may convert the first data of the RGB type into the YUV type second data.

[0028] The image signal processor 110 converts the first data into the second data and then stores the second data in the memory 300.

[0029] The shake correction module 120 may perform the shake correction of static image data or moving image data. The shake correction module 120 may perform shake correction by

reading the first data or the second data stored in the memory 300. In an embodiment, the shake correction means the detection of a shaking of the camera from the moving image data and removal of the shaking from the moving image data.

[0030] The shake correction module 120 may correct the shaking of the first data or the second data to update the first data or the second data and store the updated data in the memory 300.

[0031] The multi-format codec 130 may be a codec that compresses the moving image data. In general, since the size of the moving image data is very large, a compression module that reduces its size is necessary. The moving image data may be compressed via association among a plurality of frames, and this compression may be performed by the multi-format codec 130. The multi-format codec 130 may read and compress the first data or the second data stored in the memory 300.

[0032] The multi-format codec 130 may compress the first data or the second data to generate updated first data or updated second data and stores the updated data in the memory 300.

[0033] The GPU (Graphics Processing Unit) 140 may perform an arithmetic process to generate two-dimensional or three-dimensional graphics. The GPU 140 may arithmetically process the first data or the second data stored in the memory 300. The GPU 140 may be specialized for graphic data processing to process portions of the graphic data in parallel.

[0034] The GPU 140 may compress the first data or the second data to generate updated first data or updated second data, and stored the updated data in the memory 300.

[0035] The display 150 may display the second data stored in the memory 300 on a screen. The display 150 may display image data processed by components of the multimedia IP 100, that is, the image signal processor 110, the shake correction module 120, the multi-format codec 130 and the GPU 140. However, the present inventive concept is not limited to these examples.

[0036] The image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 may individually operate, respectively. That is, the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 may individually access the memory 300 to write or read data.

[0037] In an embodiment, the frame buffer compressor 200 compresses the second data to convert the second data into third data before the multimedia IP 100 individually accesses the memory 300. The frame buffer compressor 200 transmits the third data to the multimedia IP 100, and the multimedia IP 100 transmits the third data to the memory 300.

5 [0038] As a result, the third data compressed by the frame buffer compressor 200 is stored in the memory 300. Conversely, the third data stored in the memory 300 may be loaded by the multimedia IP 100 and transmitted to the frame buffer compressor 200. In an embodiment, the frame buffer compressor 200 decompresses the third data to convert the third data into the second data. The frame buffer compressor 200 may transmit the second data (i.e., the
10 decompressed data) to the multimedia IP 100.

[0039] In an embodiment, each time the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 individually access the memory 300, the frame buffer compressor 200 compresses the second data into the third data and transfers the third data to the memory 300. For example,
15 after one of the components of the multimedia IP 100 generates and stores the second data in the memory 300, the frame buffer compressor 200 can compress the stored data and store the compressed data into the memory 300. In an embodiment, each time a data request is transmitted from the memory 300 to the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia
20 IP, the frame buffer compressor 200 decompresses the third data into the second data, and transmits the second data to the image data processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100, respectively.

[0040] The memory 300 stores the third data generated by the frame buffer compressor 200,
25 and may provide the stored third data to the frame buffer compressor 200 so that the frame buffer compressor 200 can decompress the third data.

[0041] In an embodiment, the multimedia IP 100 and the memory 300 are connected to the system bus 400. Specifically, the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100
30 may be individually connected to the system bus 400. The system bus 400 may be a path

through which the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140, the display 150 and the memory 300 of the multimedia IP 100 transfer data to each other.

[0042] The frame buffer compressor 200 is not connected to the system bus 400, and performs the operation of converting the second data into the third data (e.g., image compression) and converting the third data into the second data (e.g., image decompression) when the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 access the memory, respectively.

[0043] Next, referring to FIG. 2, the frame buffer compressor 200 of the image processing device according to an exemplary embodiment of the present inventive concept is directly connected to the system bus 400.

[0044] The frame buffer compressor 200 is not directly connected to the multimedia IP 100 and is connected to the multimedia IP 100 via the system bus 400. Specifically, each of the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 may transmit the data to and from the frame buffer compressor 200 through the system bus 400, and may transmit the data to the memory 300 accordingly.

[0045] That is, in the process of compression, each of the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 may transmit the second data to the frame buffer compressor 200 through the system bus 400. Subsequently, the frame buffer compressor 200 may compress the second data into the third data and transmit the third data to the memory 300 via the system bus 400.

[0046] Likewise, even in the process of decompression, the frame buffer compressor 200 may receive the third data stored in the memory 300 via the system bus 400, and may decompress it to the second data. Subsequently, the frame buffer compressor 200 may transmit the second data to each of the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 via the system bus 400.

[0047] Referring to FIG. 3, in an image processing device according to an exemplary embodiment of the present inventive concept, a memory 300 and a system bus 400 are connected to each other via a frame buffer compressor 200.

[0048] That is, the memory 300 is not directly connected to the system bus 400 but is connected to the system bus 400 only via the frame buffer compressor 200. Further, the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 are directly connected to the system bus 400. Therefore, the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100 access the memory 300 only through the frame buffer compressor 200.

[0049] In the present specification, the second data is referred to as an image data 10, and the third data is referred to as compressed data 20.

[0050] FIG. 4 is a block diagram for explaining the frame buffer compressor of FIGS. 1 to 3 in detail.

[0051] Referring to FIG. 4, the frame buffer compressor 200 includes an encoder 210 (e.g., an encoding circuit) and a decoder 220 (e.g., a decoding circuit).

[0052] The encoder 210 may receive the image data 10 from the multimedia IP 100 to generate the compressed data 20. The image data 10 may be transmitted from each of the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100. The compressed data 20 may be transmitted to the memory 300 via the multimedia IP 100 and the system bus 400.

[0053] Conversely, the decoder 220 may decompress the compressed data 20 stored in the memory 300 into the image data 10. The image data 10 may be transferred to the multimedia IP 100. The image data 10 may be transmitted to each of the image signal processor 110, the shake correction module 120, the multi-format codec 130, the GPU 140 and the display 150 of the multimedia IP 100.

[0054] FIG. 5 is a block diagram for explaining the encoder of FIG. 4 in detail.

[0055] Referring to FIG. 5, the encoder 210 includes a first mode selector 219 (e.g., a logic circuit), a prediction module 211 (e.g., a logic circuit), a quantization module 213 (e.g., a logic

circuit), an entropy encoding module 215 (e.g., a logic circuit) and a padding module 217 (e.g., a logic circuit).

[0056] In an embodiment, the first mode selector 219 determines whether the encoder 210 operates in a lossless mode or a lossy mode. When the encoder 210 operates in the lossless mode in accordance with the first mode selector 219, the image data 10 is compressed along the lossless path (Lossless) of FIG. 5, and when the encoder 210 operates in the lossy mode, the image data 10 is compressed along the lossy path (Lossy).

[0057] The first mode selector 219 may receive a signal from the multimedia IP 100 which is used to determine whether to perform the lossless compression or perform the lossy compression. The lossless compression means compression without loss of data. A compression ratio may change depending on the data being losslessly compressed. Unlike lossless compression, the lossy compression is compression in which data is partly lost. The lossy compression has a higher compression ratio than the lossless compression, and may have a fixed compression ratio set in advance.

[0058] In the case of the lossless mode, the first mode selector 219 enables the image data 10 to flow along the lossless path (Lossless) to the prediction module 211, the entropy encoding module 215 and the padding module 217. Conversely, in the lossy mode, the first mode selector 219 enables the image data 10 to flow along the lossy path (Lossy) to the prediction module 211, the quantization module 213 and the entropy encoding module 215.

[0059] The prediction module 211 may compress the image data 10 by dividing the image data 10 into prediction data and residual data. The prediction data and the residual data together take up less space than the image data 10. In an embodiment, the prediction data is image data of one pixel of the image data and the residual data is created from the differences between the prediction data and the image data of the pixels of the image data that are adjacent the one pixel. For example, if the image data of the one pixel has a value between 0 and 255, 8 bits may be needed to represent this value. When the adjacent pixels have similar values to that of the one pixel, the residual data of each of the adjacent pixels is much smaller than prediction data, and thus the number of data bits representing the image data 10 may be greatly reduced. For example, when pixels having values of 253, 254, and 255 are consecutive, if the prediction data is set as 253, the residual data representation of (253 (prediction), 1 (residue), and 2 (residue))

is sufficient, and the number of bits per pixel for expressing these residual data may be decreased from 8 bits to 2 bits. For example, 24bits of data of 253, 254, and 255 can be reduced to 12bits due to 8bit prediction data of 253 (11111101), 2 bit residual data of $254-253 = 1$ (01), and 2 bit residual data of $255-253=2$ (10).

5 **[0060]** Therefore, the prediction module 211 may compress the overall size of the image data 10 by dividing the image data 10 into prediction data and residual data. Various methods are available for setting the type of the prediction data.

10 **[0061]** The prediction module 211 may perform prediction on a pixel basis or may perform prediction on a block basis. In this case, the block may mean a region formed by a plurality of adjacent pixels. For example, prediction on a pixel basis could mean that all the residual data is created from one of the pixels, and prediction on the block basis could mean that residual data is created for each block from a pixel of the corresponding block.

15 **[0062]** The quantization module 213 may further compress the image data 10 that was compressed by the prediction module 211. In an exemplary embodiment, the quantization module 213 removes the lower bits of the image data 10 through a preset quantization coefficient. Specifically, the representative value is selected by multiplying the data by the quantization coefficient, but a loss may occur by truncating the decimal part. If the value of the pixel data is between 0 and $2^8 - 1$ ($= 255$), the quantization coefficient may be defined as $/(2^n-1)$ (where, n is an integer less than or equal to 8). However, the present embodiment is not limited thereto. For example, if the prediction data is 253 (11111101), the prediction data can be reduced from 8 bits to 6 bits by removing the lower 2 bits, which results in prediction data of (111111) 252.

25 **[0063]** However, the removed lower bits are not restored later and thus are lost. Therefore, the quantization module 213 is utilized only in the lossy mode. However, since the lossy mode has a compression ratio relatively higher than that in the lossless mode and may have a fixed compression ratio set in advance, information on the compression ratio is not separately required later.

[0064] The entropy encoding module 215 may compress the image data 10 compressed by the quantization module 213 in the lossy mode or the image data 10 compressed by the prediction

module 211 in the lossless mode through entropy coding. In an embodiment, the entropy coding utilizes a method for allocating a number of bits depending on frequency.

[0065] In an exemplary embodiment, the entropy encoding module 215 compresses the image data 10, using Huffman coding. In an alternative embodiment, the entropy encoding module 215 compresses the image data 10 via exponential golomb coding or golomb rice coding. In an embodiment, the entropy encoding module 215 determines an entropy coding value (e.g., a k value) from the data it is to be compress, creates a table from the k value, and compresses the image data 10 using the table.

[0066] The padding module 217 may perform padding on the image data 10 compressed by the entropy encoding module 215 in the lossless mode. Here, the padding may mean addition of meaningless data to match a specific size. This will be explained in more detail below.

[0067] The padding module 217 may be enabled not only in the lossless mode but also in the lossy mode. In the lossy mode, the image data 10 may be compressed further than the desired compression ratio when compressed by the quantization module 213. In such a case, even in the lossy mode, the image data 10 may be converted into the compressed data 20 via the padding module 217 and transmitted to the memory 300. In an exemplary embodiment, the padding module 217 is omitted so that no padding is performed.

[0068] The compression management module 218 determines a combination of a quantization parameter (QP) table and the entropy table each used for quantization and entropy coding, and controls the compression of the image data 10 in accordance with a combination of the determined QP table and the entropy table.

[0069] In this case, the first mode selector 219 determines that the encoder 210 operates in the lossy mode, and the image data 10 is compressed along the lossy path (Lossy) of FIG. 5 accordingly. That is, the configuration in which the compression management module 218 determines the combination of the QP table and the entropy table, and compresses the image data 10 in accordance with the combination of the determined QP table and the entropy table, is premised on the case where the frame buffer compressor 200 compresses the image data 10 using a lossy compression algorithm.

[0070] Specifically, the QP table includes one or more entries, and each entry may include quantization coefficients used for quantizing the image data 10.

[0071] In an embodiment, the entropy table includes a plurality of code tables which are identified using the k value to execute the entropy coding algorithm. The entropy table which may be used in some embodiments of the present inventive concept include at least one among an exponential golomb code and a golomb rice code.

5 [0072] The compression management module 218 determines a QP table including the predetermined number of entries, and the frame buffer compressor 200 performs the quantization of the predicted image data 10, using the determined QP table. In addition, the compression management module 218 determines an entropy table, using the predetermined number of k values, and the frame buffer compressor 200 performs the entropy coding on the
10 quantized image data 10, using the determined entropy table. That is, the frame buffer compressor 200 generates the compressed data 20 on the basis of the combination of the QP table and the entropy table determined by the compression management module 218.

[0073] Thereafter, the frame buffer compressor 200 may write the generated compressed data 20 to the memory 300. Also, the frame buffer compressor 200 may read the compressed data
15 20 from the memory 300, and decompress the read compressed data 20 to generate a decompressed data to provide the decompressed data to the multimedia IP 100.

[0074] More details of the compression management module 218 for executing such operations will be described later with reference to FIGS. 7 to 16.

[0075] FIG. 6 is a block diagram for explaining the decoder of FIG. 4 in more detail.

20 [0076] Referring to FIG. 6, the decoder 220 includes a second mode selector 229 (e.g., a logic circuit), an unpadding module 227 (e.g., a logic circuit), an entropy decoding module 225 (e.g., a logic circuit), an inverse quantization module 223 (e.g., a logic circuit), and a prediction compensation module 221 (e.g., a logic circuit).

[0077] The second mode selector 229 determines whether or not the compressed data 20 stored
25 in the memory 300 has been compressed in a lossless manner or a lossy manner. In an exemplary embodiment, the second mode selector 229 determines whether the compressed data 20 has been compressed by the lossless mode or the lossy mode through the presence or absence of a header. This will be explained in more detail below.

[0078] In the case of the lossless mode, the second mode selector 229 enables the compressed
30 data 20 to flow along the lossless path (Lossless) to the unpadding module 227, the entropy

decoding module 225 and the prediction compensation module 221. Conversely, in the case of the lossy mode, the second mode selector 229 enables the compressed data 20 to flow along the lossy path (Lossy) to the entropy decoding module 225, the inverse quantization module 223 and the prediction compensation module 221.

5 **[0079]** The unpadding module 227 removes the padded portion of the data which is padded by the padding module 217 of the encoder 210. The unpadding module 227 may be omitted when the padding module 217 is omitted.

10 **[0080]** The entropy decoding module 225 may decompress the data compressed by the entropy encoding module 215. The entropy decoding module 225 may perform the decompression via Huffman coding, exponential golomb coding or golomb rice coding. Since the compressed data 20 includes the k value, the entropy decoding module 225 may perform the decoding, using the k value.

15 **[0081]** The inverse quantization module 223 may decompress the data compressed by the quantization module 213. The inverse quantization module 223 may restore the compressed data 20 compressed using the quantization coefficient determined by the quantization module 213, but it is not possible to completely restore the part which is lost in the process of compression. Therefore, the inverse quantization module 223 is utilized only in the lossy mode.

20 **[0082]** The prediction compensation module 221 may restore the data represented by the prediction data and the residual data generated by the prediction module 211. The prediction compensation module 221 may, for example, convert the residual data representation of (253 (prediction), 1 (residue), and 2 (residue)) into 253, 254, and 255. For example, the prediction compensation module 221 may restore the data by adding the residual data to the prediction data.

25 **[0083]** The prediction compensation module 221 may restore the prediction executed in units of pixels or blocks in accordance with the prediction module 211. As a result, the compressed data 20 may be restored or decompressed and may be transmitted to the multimedia IP 100.

[0084] The decompression management module 228 may perform a work in which the combination of the QP table and the entropy table determined by the compression management

module 218 described above with reference to FIG. 5 to execute compression of the image data 10 can be properly reflected when decompressing the compressed data 20.

[0085] The operation of the above-described image processing device will now be described with reference to FIGS. 7 to 16.

5 [0086] FIGS. 7 to 10 are schematic diagrams for explaining an operation of the image processing device according to an exemplary embodiment of the present inventive concept.

[0087] Referring to FIG. 7, the compression management module 218 determines a QP table 230 includes a predetermined number of entries. In this embodiment, the compression management module 218 determines the QP table 230 includes eight entries that enable
10 quantization of eight levels.

[0088] The QP table 230 includes a first entry and a second entry. Here, the first entry is a quantization coefficient corresponding to index 0, and the second entry is a quantization coefficient corresponding to index 7. Further, the QP table 230 includes one or more third entries between the first entry and the second entry. In the present embodiment, one or more
15 third entries correspond to the quantization coefficients corresponding to the indices 1 to 6.

[0089] The first entry is determined as a predetermined first value. In the present embodiment, the predetermined first value is represented by "MaxShiftValue". The first value is a constant value that may be arbitrarily determined as necessary in actual implementation of the image processing device or the method for operating the image processing device according
20 to the various embodiments of the present inventive concept.

[0090] After the first entry is determined as the first value, the second entry is determined by the following equation:

[0091] $\text{MaxShiftValue} \gg \text{BitDepth} * (1 - \text{CompressionRatio})$

[0092] Here, MaxShiftValue represents the aforementioned first value, BitDepth represents the
25 bit depth of the image data 10, and CompressionRatio represents the compression ratio.

[0093] That is, when the first values are equal and the target compression ratios are equal, the compression management module 218 may differently set the final values of the QP table 230 in accordance with the bit depth of the image data 10.

[0094] In this way, after the first entry is determined as the first value and the second entry is
30 determined as the second value, the third entries may be determined via sampling. That is, in

the QP table 230 including the predetermined number of entries, the third entries may be sampled so that the quantization coefficient is properly distributed, for the remaining third entries except the first entry and the second entry.

[0095] On the other hand, the compression management module 218 may determine the entropy table, using a predetermined number of k values. In some embodiments of the present disclosure, the entropy table is determined by a maximum of 4 or less k values.

[0096] Concretely, the entropy table may be defined by the k value of n, n+1, n+2, and n+3 (where n is an integer of 0 or more) for the image data 10 having the first bit depth, and may be defined by the k value of n+a, n+a+1, n+a+2, and n+a+3 (where, a is an integer of 1 or more) for the image data 10 having a second bit depth larger than the first bit depth.

[0097] For example, the entropy table may be determined by the k values 0, 1, 2, and 3 for the image data having the first bit depth of 8 bits, and may be determined by the k values of 1, 2, 3, and 4, for the image data having the second bit depth of 10 bits.

[0098] That is, the compression management module 218 may differently set, for example, four consecutive k values in accordance with the bit depth of the image data 10.

[0099] When performing the block-based lossy compression of the image data 10, the prediction, quantization, and entropy coding steps are performed as described above. In the case of quantization, as the QP table including the quantization factors has the more entries, it is possible to more delicately perform the compression. However, the memory resources are consumed to maintain the QP table with multiple entries, and the number of bits of the stream header of the compressed data 20 transmitted and received between the encoder 210 and the decoder 220 of the frame buffer controller 200 may increase. In particular, in an environment in which the compression is performed on the image data 10 in units of blocks, even if the header of the stream of the compressed data 20 increases only by one bit, when considering the overall image size, a considerable amount of bandwidths may be consumed.

[00100] Further, in the case of the entropy coding, since the distribution of the residue signal (residual) is different between the case of image data, for example, having the bit depth of 8 bits and the case of image data having, for example, the bit depth of 10 bits, the entropy table used for the entropy coding also needs to be determined differently. For example, since the tendency of the value of the entropy table entry to converge to 0 becomes lower in the case of

the image data having the bit depth of 10 bits than in the case of the image data having the bit depth of 8 bits, it is necessary to differently determine the k value depending on the bit depth.

[00101] Accordingly, by determining the QP table of an appropriate size with good compression quality and by selecting an appropriate k value in consideration of the distribution of the residue signal to determine the entropy table, the bandwidth between the frame buffer compressor 200 and the memory 300 can be reduced, while enhancing the compression performance of the image data 10.

[00102] In some embodiments of the present inventive concept, the compression management module 218 determines a QP table including the maximum 16 entries. Also, in some embodiments of the present disclosure, the compression management module 218 determines the QP table including eight or more entries.

[00103] Next, referring to FIG. 8, the compression management module 218 determines the QP table 232 including the eight entries.

[00104] As an example, it is assumed that the first entry of the QP table 232 is determined as 4096. The compression management module 218 performs a bit shift operation on 4096 corresponding to the first value by $\text{BitDepth} \times (1 - \text{CompressionRatio})$ as follows.

[00105] $4096 \gg 8 \times (1 - 0.5) = 256$

[00106] Here, 8 is a value corresponding to the bit depth of the image data 10, and 0.5 (e.g., 50%) is a value corresponding to the target compression ratio.

[00107] In this way, after the first entry is determined as 4096 and the second entry is determined as 256, the third entries may be determined through sampling. That is, in the QP table 232 including eight entries, the third entries may be determined, for example, as 1365, 819, 585, 455, 372, and 315 such that the quantization coefficients are properly distributed, for the third entry corresponding to the remaining indexes 1 to 6 except the first entry 4096 and the second entry 256. The entries of the QP table 242 determined thus may also be used as a quantization coefficient by being divided by, for example, 4096 which is the MaxShiftValue value.

[00108] In an embodiment, the compression management module 218 determines the k value as 0, 1, 2, and 3 when the image data 10 has a bit depth of 8 bits, and determines the k value as 1, 2, 3, and 4 when the image data 10 has the bit depth of 10 bits.

[00109] In this way, since the compression management module 218 determines a combination of the QP table and the entropy table each used for quantization and entropy coding, and controls the image data 10 in accordance with a combination of the determined QP table and the entropy table, it is possible to reduce the bandwidth between the frame buffer compressor 200 and the memory 300, while enhancing the compression performance of the image data 10.

[00110] Next, referring to FIG. 9, the compression management module 218 determines the QP table 234 including the eight entries.

[00111] For example, it is assumed that the first entry of the QP table 234 is determined as 4096. The compression management module 218 performs a bit shift operation on 4096 corresponding to the first value by $\text{BitDepth} * (1 - \text{CompressionRatio})$ as follows.

[00112] $4096 \gg 10 \times (1 - 0.5) = 128$

[00113] Here, 10 is a value corresponding to the bit depth of the image data 10, and 0.5 is a value corresponding to the target compression ratio.

[00114] In this way, after the first entry is determined as 4096 and the second entry is determined as 128, the third entries may be determined through sampling. That is, in the QP table 234 including eight entries, the third entries may be determined, for example, as 819, 455, 293, 216, 171, and 146 so that the quantization coefficients are appropriately distributed, for the third entry corresponding to the remaining indexes 1 to 6 except the first entry 4096 and the second entry 128. The entries of the QP table 242 determined in this way may also be used as a quantization coefficient by being divided by, for example, 4096 which is the MaxShiftValue value.

[00115] In an embodiment, the compression management module 218 determines the k value as 0, 1, 2, and 3 when the image data 10 has a bit depth of 8 bits, and determines the k value as 1, 2, 3, and 4 when the image data 10 has a bit depth of 10 bits.

[00116] In this way, the compression management module 218 determines a combination of the QP table and the entropy table each used for quantization and entropy coding, and controls the compression of the image data 10 in accordance with the determined combination of the QP table and the entropy table. Thus, it is possible to reduce the bandwidth between the frame buffer compressor 200 and the memory 300, while enhancing the compression performance of the image data 10.

[00117] Next, referring to FIG. 10, the compressed bit stream 236 generated from the encoder 210 of the frame buffer compressor 200 through the aforementioned process as described above includes a header including QP table information of 4 bits and k value information of 2 bits, and includes compressed data 20 as a compressed binary following the header.

5 [00118] FIGS. 11 to 14 are schematic diagrams for explaining an operation of the image processing device according to exemplary embodiments of the present disclosure.

[00119] Referring to FIG. 11, the compression management module 218 determines a QP table 240 that includes a predetermined number of entries. In this embodiment, the compression management module 218 determines the QP table 240 having 16 entries that enables
10 quantization of 16 levels.

[00120] The QP table 240 includes a first entry and a second entry. Here, the first entry is a quantization coefficient corresponding to index 0, and the second entry is a quantization coefficient corresponding to index 15. The QP table 240 includes one or more third entries between the first entry and the second entry. In the present embodiment, one or more third
15 entries correspond to the quantization coefficients corresponding to the indices 1 to 14.

[00121] After the first entry is determined as the first value and the second entry is determined as the second value, the third entries may be determined through sampling. That is, the third entry may be sampled such that the quantization coefficient is properly distributed for the remaining third entries except the first entry and the second entry, in the QP table 240 that
20 includes a predetermined number of entries.

[00122] In an embodiment, the compression management module 218 may determine the entropy table using a predetermined number of k values. In some embodiments of the present inventive concept, the entropy table may be determined by k values of maximum of 4 or less.

[00123] For example, the entropy table may be determined by k values of 0, 1, 2, and 3 for the
25 image data having the first bit depth of 8 bits, and may be determined by k values of 1, 2, 3, and 4, for the image data having a second bit depth of 10 bits.

[00124] That is, the compression management module 218 may differently set, for example, four consecutive k values in accordance with the bit depth of the image data 10.

[00125] Next, referring to FIG. 12, the compression management module 218 determines a QP
30 table 242 including 16 entries.

[00126] For example, it is assumed that the first entry of the QP table 242 is determined as 4096. The compression management module 218 performs a bit shift operation on 4096 corresponding to the first value by $\text{BitDepth} * (1 - \text{CompressionRatio})$ as follows.

[00127] $4096 \gg 8 \times (1 - 0.5) = 256$

5 [00128] Here, 8 is a value corresponding to the bit depth of the image data 10, and 0.5 is a value corresponding to the target compression ratio.

[00129] In this way, after the first entry is determined as 4096 and the second entry is determined as 256, the third entries may be determined through sampling. That is, in the QP table 232 including eight entries, the third entries may be determined, for example, as 2048,
10 1365, 1024, 819, 683, 585, 512, 455, 410, 372, 341, 315, 293, and 273 so that the quantization coefficients are appropriately distributed, for the third entries corresponding to the remaining indexes 1 to 14 except the first entry 4096 and the second entry 256. The entries of the QP table 242 determined thus may also be used as a quantization coefficient by being divided by, for example, 4096 which is a MaxShiftValue value.

15 [00130] In an embodiment, the compression management module 218 may determine the k value as 0, 1, 2, and 3 when the image data 10 has the bit depth of 8 bits, and may determine the k value as 1, 2, 3, and 4 when the image data 10 has the bit depth of 10 bits.

[00131] In this way, since the compression management module 218 determines a combination of the QP table and the entropy table each used for quantization and entropy coding, and
20 controls the compression of the image data 10 in accordance with the determined combination of the QP table and the entropy table, it is possible to reduce the bandwidth between the frame buffer compressor 200 and the memory 300, while enhancing the compression performance of the image data 10.

[00132] Next, referring now to FIG. 13, the compression management module 218 determines
25 a QP table 244 including eight entries.

[00133] Assuming that the first entry of the QP table 244 is determined as 4096. The compression management module 218 performs a bit shift operation on 4096 corresponding to the first value by $\text{BitDepth} * (1 - \text{CompressionRatio})$ as follows.

[00134] $4096 \gg 10 \times (1 - 0.5) = 128$

[00135] Here, 10 is a value corresponding to the bit depth of the image data 10, and 0.5 is a value corresponding to the target compression ratio.

[00136] In this way, after the first entry is determined as 4096 and the second entry is determined as 128, the third entries may be determined through sampling. That is, in the QP table 234 including eight entries, the third entries may be determined, for example, as 1365, 819, 585, 455, 372, 315, 273, 228, 205, 186, 171, 158, 146, and 137 so that the quantization coefficients are properly distributed, for the third entries corresponding to the remaining indexes 1 to 14 except the first entry 4096 and the second entry 128. The entries of the QP table 242 determined thus may also be used as a quantization coefficient by being dividing by, for example, 4096 which is a MaxShiftValue value.

[00137] In an embodiment, the compression management module 218 may determine the k value as 0, 1, 2, and 3 when the image data 10 has a bit depth of 8 bits, and may determine the k value as 1, 2, 3, and 4 when the image data 10 has a bit depth of 10 bits.

[00138] In this way, since the compression management module 218 determines a combination of the QP table and the entropy table each used for quantization and entropy coding, and controls the compression of the image data in accordance with the determined combination of the QP table and the entropy table, it is possible to reduce the bandwidth between the frame buffer compressor 200 and the memory 300, while enhancing the compression performance of the image data 10.

[00139] Next, referring to Fig. 14, the compressed bit stream 246 generated from the encoder 210 of the frame buffer compressor 200 through the aforementioned process as described above includes a header including QP table information of 4 bits and k value information of 2 bits, and includes the compressed data 20 as a compression binary following the header.

[00140] FIG. 15 is a diagram for explaining advantageous effects of the image processing device and the method for operating the image processing device according to some exemplary embodiments of the present inventive concept.

[00141] Referring to FIG. 15, when the number of entries of the QP table is 12 and the number of values of k is 4, the best PSNR gain is illustrated.

[00142] That is, by determining the QP table of an appropriate size with good compression quality, and by choosing an appropriate k value in consideration of the distribution of the

residue signal to determine the entropy table, the bandwidth between the frame buffer compressor 200 and the memory 300 can be reduced, while enhancing the compression performance of the image data 10.

[00143] FIG. 16 is a flowchart illustrating a method for operating an image processing device according to an exemplary embodiment of the present inventive concept.

[00144] Referring to FIG. 16, the method for operating the image processing device according to an exemplary embodiment of the present inventive concept includes determining a QP table that includes a predetermined number of entries (S1601).

[00145] In an embodiment of the present inventive concept, determination of the QP table including the predetermined number of entries includes determination of a QP table including maximum 16 entries. In another embodiment of the present inventive concept, determination of the QP table including the predetermined number of entries includes determination of the QP table including eight or more entries.

[00146] Further, the above method includes performing quantization on predicted image data, using the determined QP table (S1603).

[00147] In addition, the method includes determining an entropy table, using a predetermined number of k values (S1605).

[00148] In an embodiment of the present inventive concept, determination of the entropy table includes determination of the entropy table, using a maximum of 4 or less k values.

[00149] Further, the above method includes performing entropy coding on the quantized image data using the determined entropy table to generate compressed data (S1607).

[00150] Those skilled in the art will appreciate that many variations and modifications may be made to the exemplary embodiments without substantially departing from the principles of the present inventive concept.

What is claimed is:

1. An image processing device comprising:
a multimedia intellectual property (IP) block configured to process image data;
5 a memory; and
a frame buffer compressor (FBC) configured to compress the image data to generate compressed data and store the compressed data in the memory,

wherein the frame buffer compressor includes a logic circuit configured to determine a combination of a quantization parameter (QP) table and an entropy table and control
10 compression of the image data based on the determined combination of the QP table and the entropy table, and

wherein the QP table includes a first entry and a second entry,
the first entry is a predetermined first value, and
the second entry is determined by the following equation:

15
$$\text{MaxShiftValue} \gg \text{BitDepth} * (1 - \text{CompressionRatio})$$

where, MaxShiftValue is the first value, BitDepth is a bit depth of the image data, and CompressionRatio is a compression ratio.

2. The image processing device of claim 1, wherein the frame buffer
20 compressor writes the compressed data generated based on the combination of the QP table and the entropy table to the memory.

3. The image processing device of claim 1 or claim 2, wherein the frame buffer
compressor reads the compressed data from the memory, decompresses the read
25 compressed data to generate a decompressed data, and provides the decompressed data to the multimedia IP block.

4. The image processing device of any one of claims 1 to 3, wherein the frame
buffer compressor compresses the image data, using a lossy compression algorithm.

5. The image processing device of any one of claims 1 to 4, wherein the QP
table includes one or more third entries between the first entry and the second entry, and
the third entry is determined through sampling.

6. The image processing device of any of claims 1 to 5, wherein the entropy table is determined by a maximum of 4 or less k values for entropy encoding.

7. The image processing device of claim 6, wherein the k values includes values of n, n+1, n+2, and n+3 when the image data has a first bit depth, and

wherein the k values includes values of n+a, n+a+1, n+a+2, and n+a+3 when the image data has a second bit depth larger than the first bit depth,

wherein n is an integer ≥ 0 and a is an integer ≥ 1 .

8. An image processing device comprising:

a multimedia intellectual property (IP) block configured to process image data;

a memory; and

a frame buffer compressor (FBC) configured to compress the image data to generate compressed data and store the compressed data in the memory,

wherein the frame buffer compressor includes a logic circuit configured to determine a quantization parameter (QP) table including a maximum of 16 entries and an entropy table determined by a maximum of 4 k values for entropy encoding, and control compression of the image data in accordance with the determined combination of the QP table and the entropy table, and

wherein the QP table includes a first entry and a second entry,

the first entry is a predetermined first value, and

the second entry is determined by the following equation:

$$\text{MaxShiftValue} \gg \text{BitDepth} * (1 - \text{CompressionRatio})$$

where, MaxShiftValue is the first value, BitDepth is a bit depth of the image data, and CompressionRatio is a compression ratio.

9. The image processing device of claim 8, wherein the QP table includes one or more third entries between the first entry and the second entry,

wherein the third entry is determined through sampling.

10. The image processing device of claim 8 or claim 9, wherein the k values include values of n, n+1, n+2, and n+3 when the image data has a first bit depth, and

wherein the k values includes values of n+a, n+a+1, n+a+2, and n+a+3 when the image data has a second bit depth larger than the first bit depth,

wherein n is an integer ≥ 0 and a is an integer ≥ 1 .

5 11. A method for operating an image processing device, the method comprising:
converting image data into predicted image data comprising prediction data and residual data;

determining a quantization parameter (QP) table including a predetermined number of entries;

10 quantizing the predicted image data to generate quantized image data, using the determined QP table;

determining an entropy table, using a predetermined number of k values for an entropy encoding; and

15 performing the entropy coding on the quantized image data, using the determined entropy table, to generate compressed data,

wherein the QP table includes a first entry and a second entry, and

the determining of the QP table including the predetermined number of entries comprises:

determining the first entry as a predetermined first value, and

20 determining the second entry by the following equation:

$$\text{MaxShiftValue} \gg \text{BitDepth} * (1 - \text{CompressionRatio})$$

where, MaxShiftValue is the first value, BitDepth is a bit depth of the image data, and CompressionRatio is a compression ratio.

25 12. The method of claim 11, wherein the determining of the QP table including the predetermined number of entries comprises:

determining a QP table including a maximum of 16 entries.

30 13. The method of claim 12, wherein determining of the QP table including the predetermined number of entries comprises:

determining a QP table including eight or more entries.

14. The method of any one of claims 11 to 13, wherein the QP table includes one or more third entries between the first entry and the second entry, and

determining of the QP table including the predetermined number of entries comprises:

5 determining the third entry through sampling.

15. The method of any one of claims 11 to 14, wherein the determining of the entropy table comprises:

10 determining the entropy table, using a maximum of 4 or less k values for entropy encoding.

16. The method of any one of claims 11 to 15, wherein the determining of the entropy table comprises:

15 determining the entropy table using the k values of n, n+1, n+2, and n+3 when the image data has a first bit depth, and

determining the entropy table using the k values of n+a, n+a+1, n+a+2, and n+a+3 when the image data has a second bit depth larger than the first bit depth,

wherein n is an integer ≥ 0 , and

wherein a is an integer ≥ 1 .

20

17. The method of any one of claims 11 to 16, wherein the entropy table includes at least one of an exponential golomb code and a golomb rice code.