DESCRIPTION

Title of Invention:

IMAGE SIGNAL DECODING APPARATUS, IMAGE SIGNAL DECODING METHOD, IMAGE SIGNAL ENCODING APPARATUS, IMAGE SIGNAL ENCODING METHOD, AND PROGRAM

Technical Field

[0001]

The present invention relates to an image signal decoding apparatus, an image signal decoding method, an image signal encoding apparatus, an image signal encoding method, and a program, and more specifically, to an image signal decoding apparatus, an image signal decoding method, an image signal encoding apparatus, an image signal encoding method, and a program that allow easy detection of picture boundaries of a Dependent stream on a bit stream.

Background Art

[0002]

In recent years, apparatuses complying with schemes for compression based on an orthogonal transform such as a discrete cosine transform and motion compensation, such as MPEG, by utilizing redundancy specific to image information for the purpose of realizing high-efficiency transmission and accumulation of information have been increasingly

prevalent for use in both distribution of information from broadcast stations or the like and receipt of information at general consumer homes.

[0003]

In particular, MPEG2 (ISO/IEC 13818-2) is defined as a general-purpose image encoding scheme, and is a standard that covers both interlaced scanned images and progressive scanned images as well as standard-definition images and high-definition images, which is now widely used for a wide variety of applications including professional applications and consumer applications.

[0004]

With the use of MPEG2, a high compression ratio and high quality implementation of images is feasible by, for example, assigning a code rate (bit rate) 4 to 8 Mbps to a standard-definition interlaced scanned image having 720×480 pixels. In addition, a high compression ratio and high quality implementation of images is feasible by assigning a code rate of 18 to 22 Mbps to a high-definition interlaced scanned image having 1920×1088 pixels.

[0005]

MPEG2 has been intended primarily for high-quality encoding suitable for broadcasting, but has not supported encoding schemes of a lower code rate, that is, a higher compression ratio, than that of MPEG1.

[0006]

With the increase in popularity of mobile terminals, the demand for such encoding schemes will increase in the future. To meet the demand, the MPEG4 encoding scheme was standardized. As to image encoding schemes, the ISO/IEC 14496-2 standard was approved as an international standard in December 1998.

[0007]

Furthermore, a standard called H.264/AVC (MPEG-4 part 10, ISO/IEC 14496-10 | ITU-T H.264) is also standardized. The standard is developed by an organization named JVT (Joint Video Team) jointly established by ITU-T and ISO/IEC to promote standardization of video encoding.

[0008]

It is known that H.264/AVC requires a larger amount of computation for its encoding and decoding than conventional encoding schemes such as MPEG2 and MPEG4, but makes a higher encoding efficiency feasible.

[0009]

[H.264/AVC]

Fig. 1 is a block diagram illustrating an example configuration of an image information encoding apparatus that implements image compression based on an orthogonal transform such as a discrete cosine transform or a Karhunen-Loève transform and motion compensation.

[0010]

1 denotes an A/D conversion unit, 2 denotes a screen rearrangement buffer, 3 denotes an adder unit, 4 denotes an orthogonal transform unit, 5 denotes a quantization unit, 6 denotes a lossless encoding unit, 7 denotes an accumulation buffer, 8 denotes a dequantization unit, 9 denotes an inverse orthogonal transform unit, 10 denotes a frame memory, 11 denotes a motion prediction/compensation unit, and 12 denotes a rate control unit.

[0011]

An image signal that is input is first converted into a digital signal by the A/D conversion unit 1.

[0012]

Then, frames are rearranged by the screen rearrangement buffer 2 in accordance with the GOP (Group of Pictures) structure of image compression information that is output.

[0013]

For an image to be subjected to intra-encoding, image information about the entire frames is input to the orthogonal transform unit 4, where an orthogonal transform such as a discrete cosine transform or a Karhunen-Loève transform is performed.

[0014]

A transform coefficient that is the output of the orthogonal transform coefficient 4 is subjected to

quantization processing by the quantization unit 5.

[0015]

A quantized transform coefficient that is the output of the quantization unit 5 is input to the lossless encoding unit 6, where lossless coding such as variable length coding or arithmetic coding is performed. Thereafter, the resulting transform coefficient is accumulated in the accumulation buffer 7, and is output as image compression information. The behavior operation of the quantization unit 5 is controlled by the rate control unit 12.

[0016]

Simultaneously, the quantized transform coefficient that is the output of the quantization unit 5 is input to the dequantization unit 8, and is in addition subjected to inverse orthogonal transform processing by the inverse orthogonal transform unit 9 into decoded image information. The information is accumulated in the frame memory 10.

[0017]

An image to be subjected to inter-encoding is first input from the screen rearrangement buffer 2 to the motion prediction/compensation unit 11.

[0018]

Simultaneously, image information to be referred to is retrieved from the frame memory 10, and is subjected to motion prediction/compensation processing. Reference image

information is generated.

[0019]

The reference image information is sent to the adder unit 3, and is converted here into a difference signal between the reference image information and the image information.

[0020]

The motion compensation/prediction unit 11 simultaneously outputs motion vector information to the lossless encoding unit 6. The motion vector information is subjected to lossless coding processing such as variable length coding or arithmetic coding, and is inserted in the header portion of the image compression information. Other processing is similar to that for the image compression information to be subjected to intra-encoding.

[0021]

Fig. 2 is a block diagram illustrating an example configuration of an image information decoding apparatus.

[0022]

21 denotes an accumulation buffer, 22 denotes a lossless encoding/decoding unit, 23 denotes a dequantization unit, 24 denotes an inverse orthogonal transform unit, 25 denotes an adder unit, 26 denotes a screen rearrangement buffer, 27 denotes a D/A conversion unit, 28 denotes a frame memory, and 29 denotes a motion prediction/compensation unit.

[0023]

Image compression information (bit stream) that is input is first stored in the accumulation buffer 21, and is thereafter transferred to the lossless encoding/decoding unit 22.

[0024]

In the lossless encoding/decoding unit 22, processing such as variable length decoding or arithmetic decoding is performed in accordance with a determined image compression information format.

[0025]

Simultaneously, if the frame is an inter-encoded frame, the lossless encoding/decoding unit 22 also decodes motion vector information stored in the header portion of the image compression information, and outputs the information to the motion prediction/compensation unit 29.

[0026]

A quantized transform coefficient that is the output of the lossless encoding/decoding unit 22 is input to the dequantization unit 23, and is here output as a transform coefficient.

[0027]

The transform coefficient is subjected to an inverse orthogonal transform such as an inverse discrete cosine transform or an inverse Karhunen-Loève transform by the

inverse orthogonal transform unit 24 in accordance with a determined image compression information format.

[0028]

In a case where the frame is an intra-encoded frame, image information subjected to inverse orthogonal transform processing is stored in the screen rearrangement buffer 26, and is output after D/A conversion processing.

[0029]

In a case where the frame is an inter-encoded frame, a reference image is generated based on the motion vector information subjected to lossless decoding process and the image information stored in the frame memory 28. The reference image and the output of the inverse orthogonal transform unit 24 are combined by the adder unit 25. Other processing is similar to that for the intra-encoded frame.

[0030]

The AVC standard developed by the JVT described previously is a hybrid coding scheme formed of motion compensation and a discrete cosine transform, like MPEG2 or MPEG4.

[0031]

A discrete cosine transform may be an integer transform approximating a real discrete cosine transform. Although detailed schemes are different such as the transform method of the discrete cosine transform being a method that uses an

integer coefficient with a 4×4 block size or the block size in motion compensation being variable, the basic scheme is similar to that of the encoding scheme implemented with the configuration in Fig. 1.

[0032]

Meanwhile, in recent years, with the advancement of stereoscopic image capture and display technologies, studies on an extension of H.264/AVC to encoding of stereoscopic image signals have been advanced.

[0033]

Standardization of MVC (Multiview Video Coding) that allows encoding of multi-viewpoint images captured using a plurality of image capture apparatuses is developed.

[0034]

An image that is assumed to be captured and displayed from two viewpoints is called a stereo image. A naked-eye stereo display is capable of supporting multi-viewpoint display.

[0035]

While the following description will be given of, mainly, a two-viewpoint stereo image by way of example, application to multi-viewpoint images obtained from three or more viewpoints in a similar manner can be made.

[0036]

[MVC]

Fig. 3 is a diagram illustrating a multi-viewpoint encoding apparatus.

[0037]

In a multi-viewpoint encoding apparatus 41, video signals supplied from two image capture apparatuses, that is, image capture apparatuses 31 and 32, are encoded, and bit streams generated by encoding are output. The bit streams composed of data of two-viewpoint images may be multiplexed into a single stream which is output, or may be output as two or more bit streams.

[0038]

Fig. 4 is a block diagram illustrating an example configuration of the multi-viewpoint encoding apparatus 41 in Fig. 3.

[0039]

In the multi-viewpoint encoding apparatus 41, a one-viewpoint image among multi-viewpoint images is encoded as a Base stream, and the other images are encoded as Dependent streams.

[0040]

In the case of a stereo image, one image out of an L image (left-viewpoint image) and an R image (right-viewpoint image) is encoded as a Base stream, and the other image is encoded as a Dependent stream.

[0041]

The Base stream is a bit stream similar to an existing AVC bit stream encoded using H.264 AVC/High Profile or the like. Therefore, the Base stream becomes a stream that can be decoded using an existing AVC decoder supporting H.264 AVC/High Profile.

[0042]

Images to be encoded as a Base stream are input to a rearrangement buffer 51, and are rearranged in an order suitable for encoding as I pictures, P pictures, and B pictures. The rearranged images are output to a video encoding unit 52.

[0043]

The video encoding unit 52 has a similar configuration to the image information encoding apparatus in Fig. 1. In the video encoding unit 52, for example, encoding is performed in compliance with H.264 AVC/High Profile, and a resulting bit stream is output to a multiplexing unit 57. In addition, a local decoded image is saved in a frame memory 53, and is used as a reference image for encoding the next picture or a picture in the Dependent stream.

[0044]

In the meantime, images to be encoded as a Dependent stream are input to a rearrangement buffer 54, and are rearranged in an order suitable for encoding as I pictures, P pictures, and B pictures. The rearranged images are

output to a dependent stream encoding unit 55.

[0045]

In the dependent stream encoding unit 55, in addition to normal AVC encoding, encoding using, as a reference image, a local decoded image in the Base stream stored in a frame memory 53 is performed, and a bit stream is output to the multiplexing unit 57. In addition, the local decoded image is saved in the frame memory 56, and is used as a reference image for encoding the next picture.

[0046]

In the multiplexing unit 57, the Base stream and the Dependent stream are multiplexed into a single bit stream which is output. The Base stream and the Dependent stream may be output as separate bit streams.

[0047]

Fig. 5 is a diagram illustrating an example of an MVC reference image.

[0048]

A Base stream is encoded by performing only prediction in the time direction in a manner similar to that in normal AVC.

[0049]

A Dependent stream is encoded by performing, in addition to prediction in the time direction within a same-viewpoint image, which is similar to that in normal AVC,

prediction using an image in the Base stream that is obtained at the same time point as a reference image. Even in a case where prediction in the time direction cannot be suitably performed, the capability of referring to an other-viewpoint image obtained at the same time point can improve encoding efficiency.

[0050]

Fig. 6 is a block diagram illustrating the configuration of the video encoding unit 52 in Fig. 4 that generates a Base stream, and the frame memory 53.

[0051]

The configuration illustrated in Fig. 6 is similar to the configuration of the image information encoding apparatus in Fig. 1, except for the point that an image saved in the frame memory 53 is referred to by the dependent stream encoding unit 55.

[0052]

Fig. 7 is a block diagram illustrating the configuration of the dependent stream encoding unit 55 in Fig. 4 that generates a Dependent stream, and the frame memory 56.

[0053]

The configuration illustrated in Fig. 7 is similar to the configuration of the image information encoding apparatus in Fig. 1, except for the point that an image

saved in the frame memory 53 can be referred to. A reference image read from the frame memory 53 is input to a motion prediction/compensation unit 90, and is used for motion prediction and motion compensation.

[0054]

Fig. 8 is a block diagram illustrating an example configuration of a multi-viewpoint decoding apparatus 101.

[0055]

A Base stream supplied from the multi-viewpoint encoding apparatus 41 via a network or a recording medium is input to a buffer 111, and a Dependent stream is input to a buffer 114. In a case where a single multiplexed stream is supplied, the stream is separated into a Base stream and a Dependent stream which are input to the buffer 111 and the buffer 114, respectively.

[0056]

The Base stream which is delayed in the buffer 111 for a predetermined period of time is output to a video decoding unit 112.

[0057]

In the video decoding unit 112, the Base stream is decoded in accordance with AVC, and a resulting decoded image is saved in a frame memory 113. The decoded image saved in the frame memory 113 is used as a reference image for decoding the next picture or a picture in the Dependent

stream.

[0058]

The decoded image obtained by the video decoding unit 112 is output as a video signal to a 3D display 102 at a predetermined timing.

[0059]

In the meantime, the Dependent stream which is delayed in the buffer 114 for a predetermined period of time is output to a dependent stream decoding unit 115.

[0060]

In the dependent stream decoding unit 115, the Dependent stream is decoded, and a resulting decoded image is saved in a frame memory 116. The decoded image saved in the frame memory 116 is used as a reference image for decoding the next picture.

[0061]

In the dependent stream decoding unit 115, as appropriate, the image saved in the frame memory 113 is used as a reference image in accordance with information (such as a flag) in the bit stream.

[0062]

The decoded image obtained by the dependent stream decoding unit 115 is output as a video signal to the 3D display 102 at a predetermined timing.

[0063]

In the 3D display 102, a stereo image is displayed in accordance with the video signal supplied from the video decoding unit 112 and the video signal supplied from the dependent stream decoding unit 115.

[0064]

Fig. 9 is a diagram illustrating the configuration of the video decoding unit 112 in Fig. 8 that decodes a Base stream, and the frame memory 113.

[0065]

The configuration illustrated in Fig. 9 is similar to the configuration of the image information decoding apparatus in Fig. 2, except for the point that the image saved in the frame memory 113 is referred to by the dependent stream decoding unit 115.

[0066]

Fig. 10 is a block diagram illustrating the configuration of the dependent stream decoding unit 115 in Fig. 8 that decodes a Dependent stream, and the frame memory 116.

[0067]

The configuration illustrated in Fig. 10 is similar to the configuration of the image information decoding apparatus in Fig. 2, except for the point that the image saved in the frame memory 113 can be referred to. A reference image read from the frame memory 113 is input to a

motion prediction/compensation unit 148, and is used for motion prediction and motion compensation.

Citation List

Patent Literature

[0068]

PTL 1: Japanese Unexamined Patent Application Publication No. 2007-208917

Summary of Invention

Technical Problem

[0069]

In AVC and MVC, a virtual buffer model that simulates the operation of a buffer on the decoding apparatus side is defined in order to prevent buffer overflow or underflow.

[0070]

On the encoding apparatus side, encoding is performed so as to prevent the virtual buffer from overflowing or underflowing. A decoding apparatus can decode the bit stream encoded in this manner without breaking a buffer.

[0071]

A virtual buffer model will be described.

[0072]

Fig. 11 is a diagram illustrating an example of a virtual buffer model.

[0073]

An input bit stream is first supplied to a buffer 151.

The bit stream is read instantaneously from the buffer 151 in units called AUs (access units), and is supplied to a video decoding unit 152.

[0074]

In the video decoding unit 152, data read from the buffer 151 is decoded instantaneously and a decoded image is generated. The above assumption is made in the virtual buffer model.

[0075]

It is to be noted that the buffer 151 in Fig. 11 corresponds to the buffers 111 and 114 in the multi-viewpoint decoding apparatus 101 in Fig. 8. In addition, the video decoding unit 152 in Fig. 11 corresponds to the video decoding unit 112 and the dependent stream decoding unit 115 in the multi-viewpoint decoding apparatus 101 in Fig. 8.

[0076]

In the case of AVC, one AU stores data of one picture.

[0077]

In the case of MVC, one AU stores data of all views (the Base stream and the Dependent stream) obtained at the same time point. Here, the term view means an image obtained from each viewpoint.

[0078]

The behavior of the buffer 151 in such a virtual buffer

model is illustrated in Fig. 12. In Fig. 12, the vertical axis represents the amount of buffer occupancy, and the horizontal axis represents time.

[0079]

As illustrated in Fig. 12, a bit stream is supplied to the buffer 151 at a predetermined bit rate. In addition, at the read time of each AU, the bit stream stored in the corresponding AU is extracted from the buffer 151. Times t_1, t_2, t_3, ... are read times from the buffer 151.

[0080]

Fig. 13 is a diagram illustrating the configuration of AUs of a bit stream encoded using MVC.

[0081]

In the upper part of Fig. 13, AUs #1, #2, and #3 are illustrated.

[0082]

The AU #1 is composed of a picture P_1 in the Base stream and a picture P_2 in the Dependent stream, which are pictures at the same time point (data of the pictures is stored).

[0083]

The AU #2 is composed of a picture P_3 in the Base stream and a picture P_4 in the Dependent stream, which are pictures at the same time point.

[0084]

The AU #3 is composed of a picture P_5 in the Base stream

and a picture P_6 in the Dependent stream, which are pictures at the same time point.

[0085]

In the lower part of Fig. 13, the configuration of a bit stream is illustrated.

[0086]

The bit stream in Fig. 13 is a single stream configured by multiplexing the Base stream and the Dependent stream. Each AU has a NAL unit called AD (AU Delimiter) inserted at the beginning thereof. The AD is a certain unique data string.

[0087]

The AD represents the start position of data of a picture in the Base stream and a picture in the Dependent stream which are at the same time point.

[0088]

As illustrated in Fig. 13, in the bit stream, the AU #1 is composed of AD #1, picture P_1, and picture P_2.

[0089]

The AU #2 is composed of AD #2, picture P_3, and picture P_4.

[0090]

The AU #3 is composed of AD #3, picture P_5, and picture P_6.

[0091]

Searching for an AD makes it possible to find the beginning of the AU and to easily access a predetermined picture.

[0092]

Fig. 14 includes diagrams illustrating an example of the structure of bit streams.

[0093]

Part A of Fig. 14 illustrates a structure in a case where of a Base stream and a Dependent stream are multiplexed into a single bit stream. The structure of the bit stream in part A of Fig. 14 is the same as the structure of the bit stream in Fig. 13.

[0094]

Part B of Fig. 14 illustrates a structure in a case where a Base stream and a Dependent stream are separately included in a total of two bit streams. Portions corresponding to the portions illustrated in part A of Fig. 14 are assigned the same symbols.

[0095]

As illustrated in part B of Fig. 14, the Base stream is configured by arranging AD #1, picture P_1, AD #2, picture P_3, the AD #3, and picture P_5 in this order.

[0096]

In the meantime, the Dependent stream is configured by arranging picture P_2, picture P_4, and picture P_6 in this

order.

[0097]

A decoding process for the bit streams in Fig. 14 will be described using the configuration in Fig. 11.

[0098]

For example, in a case where the bit stream in part A of Fig. 14 is supplied, after detecting the AD #1, the video decoding unit 152 in Fig. 11 sequentially reads the picture P_1 in the Base stream and the picture P_2 in the Dependent stream, and individually decodes the pictures.

[0099]

In addition, in a case where the bit streams in part B of Fig. 14 are supplied, for the first AU in the Base stream, the video decoding unit 152 detects the AD #1, and reads and decodes the picture P_1. In addition, for the second AU, the video decoding unit 152 detects the AD #2, and reads and decodes the picture P_3.

[0100]

Since the Dependent stream contains no ADs, it is necessary to analyze the syntax in the bit stream and to determine the boundaries of the pictures P_2, P_4, and P_6 in order to read each picture. Therefore, the processing is very complicated.

[0101]

Here, a case is considered where an AD is simply added

to the beginning of each picture in the Dependent stream in a manner similar to that in the Base stream so that the beginning of the picture can be readily detected.

[0102]

In this case, for example, if the Base stream and the Dependent stream are multiplexed into a single bit stream, the video decoding unit 152 may recognize an AU of the Base stream and an AU of the Dependent stream as different AUs. This does not make it possible to correctly reproduce the operation of the virtual buffer model.

[0103]

Embodiments of the present invention allow easy detection of picture boundaries in a Dependent stream on a bit stream.

Solution to Problem

[0104]

According to a first aspect of the present invention, there is provided an image signal decoding apparatus that decodes a bit stream produced by encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, including decoding means for decoding a first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint,

identifying the start of encoded data obtained at the predetermined time point, decoding a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint, identifying the start of encoded data of an image signal obtained from the other viewpoint, and decoding the bit stream.

[0105]

The image signals obtained from the plurality of viewpoints can be image signals obtained from two viewpoints, and are configured as a single bit stream.

[0106]

Image signals obtained from a plurality of viewpoints can be image signals obtained from two viewpoints, and are configured as two bit streams.

[0107]

The decoding means can be configured to decode the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at the predetermined time point, the first unique data string being encoded in an image obtained from one of the viewpoints, identify the start of encoded data obtained at the predetermined time point, decode the second unique data

string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point, the second unique data string being encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint, identify the start of encoded data of an image signal obtained from the other viewpoint, and decode the bit stream.

[0108]

In the case of accessing a position at which decoding can be started correctly, the decoding means can be configured to decode the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at the predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint, identify the start of encoded data obtained at the predetermined time point from the one viewpoint, decode the second unique data strings indicating the start of the image encoded bit stream obtained from the other viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of the encoded bit stream of the image obtained from the other viewpoint, identify the start of the encoded data of the image signal obtained from the other viewpoint, and decode the bit stream from identified start positions.

[0109]

According to a second aspect of the present invention, there is provided an image signal decoding method for decoding a bit stream produced by encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, including the steps of decoding a first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint; identifying the start of encoded data obtained at the predetermined time point; decoding a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint; identifying the start of encoded data of an image signal obtained from the other viewpoint; and decoding the bit stream.

[0110]

According to a third aspect of the present invention, there is provided a program for causing a computer to execute a process for decoding a bit stream produced by encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, the program causing the computer to execute a process comprising the steps of decoding a first

unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint; identifying the start of encoded data obtained at the predetermined time point; decoding a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint; identifying the start of encoded data of an image signal obtained from the other viewpoint; and decoding the bit stream.

[0111]

According to a fourth aspect of the present invention, there is provided an image signal encoding apparatus that encodes a stereoscopic image signal including image signals obtained from a plurality of viewpoints, including encoding means for encoding a first unique data string in an image obtained from one viewpoint, the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, and individually encoding a second unique data string at the beginning of an encoded bit stream of an image obtained from another viewpoint, the second unique data

string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point.

[0112]

The encoding means can be configured to multiplex encoded data obtained from all the viewpoints into a single bit stream.

[0113]

The encoding means can be configured to cause encoded data obtained from all the viewpoints to be included in two bit streams.

[0114]

The image signals obtained from the plurality of viewpoints can be image signals obtained from two viewpoints.

[0115]

According to a fifth aspect of the present invention, there is provided an image signal encoding method for encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, including the steps of encoding a first unique data string in an image obtained from one viewpoint, the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point; and individually encoding a second unique data string at the beginning of an encoded bit stream of an image obtained from another

viewpoint, the second unique data string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point.

[0116]

According to a sixth aspect of the present invention, there is provided a program for causing a computer to execute a process for encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, the program causing the computer to execute a process comprising the steps of encoding a first unique data string in an image obtained from one viewpoint, the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point; and individually encoding a second unique data string at the beginning of an encoded bit stream of an image obtained from another viewpoint, the second unique data string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point.

[0117]

In an embodiment of the present invention, a first unique data string indicating the start of image encoded bit streams obtained from all viewpoints at a predetermined time point, which is encoded in an image obtained from one viewpoint, is decoded, and the start of encoded data obtained at the predetermined time point is identified. In

addition, a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point, which is individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint, is decoded, the start of encoded data of an image signal obtained from the other viewpoint is identified, and a bit stream is decoded.

[0118]

In another embodiment of the present invention, a first unique data string indicating the start of image encoded bit streams obtained from all viewpoints at a predetermined time point is encoded in an image obtained from one viewpoint, and a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point is individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint.

Advantageous Effects of Invention

[0119]

According to the above aspects of the present invention, it is possible to easily detect picture boundaries of a Dependent stream on a bit stream.

Brief Description of Drawings

[0120]

[Fig. 1] Fig. 1 is a block diagram illustrating an example configuration of an image information encoding apparatus.

[Fig. 2] Fig. 2 is a block diagram illustrating an example configuration of an image information decoding apparatus.

[Fig. 3] Fig. 3 is a diagram illustrating a multi-viewpoint encoding apparatus.

[Fig. 4] Fig. 4 is a block diagram illustrating an example configuration of the multi-viewpoint encoding apparatus in Fig. 3.

[Fig. 5] Fig. 5 is a diagram illustrating an example of an MVC reference image.

[Fig. 6] Fig. 6 is a block diagram illustrating the configuration of a video encoding unit in Fig. 4 and a frame memory.

[Fig. 7] Fig. 7 is a block diagram illustrating the configuration of a dependent stream encoding unit in Fig. 4 and a frame memory.

[Fig. 8] Fig. 8 is a block diagram illustrating an example configuration of a multi-viewpoint decoding apparatus.

[Fig. 9] Fig. 9 is a diagram illustrating the configuration of a video decoding unit in Fig. 8 and a frame memory.

[Fig. 10] Fig. 10 is a block diagram illustrating the configuration of a dependent stream decoding unit in Fig. 8 and a frame memory.

[Fig. 11] Fig. 11 is a diagram illustrating an example of a virtual buffer model.

[Fig. 12] Fig. 12 is a diagram illustrating the behavior of a buffer in Fig. 11.

[Fig. 13] Fig. 13 is a diagram illustrating the configuration of AUs of a bit stream encoded using MVC.

[Fig. 14] Fig. 14 includes diagrams illustrating an example of the structure of bit streams.

[Fig. 15] Fig. 15 is a block diagram illustrating an example configuration of a dependent stream encoding unit that is a portion of a multi-viewpoint encoding apparatus according to an embodiment of the present invention.

[Fig. 16] Fig. 16 includes diagrams illustrating an example of the structure of bit streams.

[Fig. 17] Fig. 17 is a diagram illustrating an example configuration of a NAL unit.

[Fig. 18] Fig. 18 is a diagram illustrating the syntax of the NAL unit.

[Fig. 19] Fig. 19 is a diagram illustrating the definition of nal_unit_type.

[Fig. 20] Fig. 20 is a diagram illustrating the definition of nal_unit_type.

[Fig. 21] Fig. 21 is a diagram illustrating the simplified syntax of a NAL unit.

[Fig. 22] Fig. 22 is a diagram illustrating syntax including the NAL unit.

[Fig. 23] Fig. 23 is a diagram illustrating a simplification of the syntax in Fig. 22.

[Fig. 24] Fig. 24 is a block diagram illustrating an example configuration of a dependent stream decoding unit that is a portion of a multi-viewpoint decoding apparatus according to an embodiment of the present invention.

[Fig. 25] Fig. 25 is a diagram illustrating an example of a virtual buffer model.

[Fig. 26] Fig. 26 is a diagram illustrating an example of a method for accessing the beginning of an AU.

[Fig. 27] Fig. 27 is a flowchart describing the operation of a syntax analysis unit.

[Fig. 28] Fig. 28 is a flowchart describing a conventional analysis process.

[Fig. 29] Fig. 29 is a diagram illustrating the configuration of AUs of bit streams obtained by encoding three-viewpoint video using MVC.

[Fig. 30] Fig. 30 includes diagrams illustrating an example of the structure of bit streams.

[Fig. 31] Fig. 31 includes diagrams illustrating an example of the structure of bit streams.

[Fig. 32] Fig. 32 is a block diagram illustrating an example configuration of a computer.

Description of Embodiments

[0121]

[Configuration of Multi-Viewpoint Encoding Apparatus]

Fig. 15 is a block diagram illustrating an example configuration of a dependent stream encoding unit that is a portion of a multi-viewpoint encoding apparatus according to an embodiment of the present invention.

[0122]

In a multi-viewpoint encoding apparatus according to an embodiment of the present invention, a Base stream has encoded therein information indicating AU boundaries, and a Dependent stream has encoded therein information indicating boundaries between pictures in the Dependent stream.

[0123]

This enables a multi-viewpoint decoding apparatus to, on the occasion of decoding bit streams of a stereoscopic image, determine AU boundaries or the like with easy processing and to perform the decoding operation based on a virtual buffer model.

[0124]

The multi-viewpoint encoding apparatus has the same configuration as the configuration in Fig. 4. Fig. 15 illustrates an example configuration of the dependent stream

encoding unit 55 in Fig. 4. A video encoding unit 52 of a multi-viewpoint encoding apparatus 41 according to an embodiment of the present invention has the same configuration as the configuration in Fig. 6. The same configuration elements are assigned the same numerals.

[0125]

The configuration of the dependent stream encoding unit 55 illustrated in Fig. 15 is the same configuration as the configuration in Fig. 7, except for the point that a DD encoding unit 201 is provided and the point that the output of the DD encoding unit 201 is supplied to the lossless encoding unit 86.

[0126]

The DD encoding unit 201 inserts a NAL unit called DD (Dependent delimiter) (View and dependency representation delimiter NAL unit) at the beginning of each picture in a Dependent stream. The DD is a unique data string like the AD but has a value different from the AD.

[0127]

The lossless encoding unit 86 inserts the NAL unit of the DD generated by the DD encoding unit 201 into the beginning of each picture, performs encoding, and outputs a Dependent stream into which the DD has been inserted.

[0128]

Fig. 16 includes diagrams illustrating an example of

the structure of bit streams generated by the multi-viewpoint encoding apparatus 41 including the dependent stream encoding unit 55 in Fig. 15.

[0129]

Part A of Fig. 16 illustrates an example of a structure in a case where a Base stream and a Dependent stream are multiplexed into a single bit stream.

[0130]

Part B of Fig. 16 illustrates an example of a structure in a case where a Base stream and a Dependent stream are separately included in a total of two bit streams. Portions corresponding to the portions illustrated in part A of Fig. 16 are assigned the same symbols.

[0131]

In the case of generating the bit stream in part A of Fig. 16, after encoding the AD #1, the multi-viewpoint encoding apparatus 41 encodes the picture P_1 in the Base stream. Next, the multi-viewpoint encoding apparatus 41 encodes DD #1 before encoding the pictures in the Dependent stream, and thereafter encodes the picture P_2 in the Dependent stream.

[0132]

When the process target is switched to the next AU, the multi-viewpoint encoding apparatus 41 encodes the AD #2 and encodes the picture P_3 in the Base stream. Next, the multi-

viewpoint encoding apparatus 41 encodes DD #2 before encoding the pictures in the Dependent stream, and thereafter encodes the picture P_4 in the Dependent stream.

[0133]

In a similar manner, the multi-viewpoint encoding apparatus 41 encodes DDs before encoding the pictures in the Dependent stream, and arranges the pictures in the Dependent stream immediately after the DDs.

[0134]

Also in this example, ADs are placed at the beginning of AUs. The first AU is composed of AD #1, picture P_1, DD #1, and picture P_2, and the second AU is composed of AD #2, picture P_3, DD #2, and picture P_4. The third AU is composed of AD #3, picture P_5, DD #3, and picture P_6.

[0135]

In the case of generating the bit streams in part B of Fig. 16, for the Base stream, after encoding the AD #1, the multi-viewpoint encoding apparatus 41 encodes the picture P_1 in the Base stream. When the process target is switched to the next AU, the multi-viewpoint encoding apparatus 41 encodes the AD #2, and encodes the picture P_3 in the Base stream.

[0136]

In addition, for the Dependent stream, after encoding the DD #1, the multi-viewpoint encoding apparatus 41 encodes

the picture P_2 in the Dependent stream. When the process target is switched to the next AU, the multi-viewpoint encoding apparatus 41 encodes the DD #2 and encodes the picture P_4 in the Dependent stream.

[0137]

In a similar manner, the multi-viewpoint encoding apparatus 41 encodes the DDs before encoding the pictures in the Dependent stream, and arranges the pictures in the Dependent stream immediately after the DDs.

[0138]

As illustrated in part B of Fig. 16, the Base stream is configured by arranging AD #1, picture P_1, AD #2, picture P_3, AD #3, and picture P_5 in this order.

[0139]

In the meantime, the Dependent stream is configured by arranging DD #1, picture P_2, DD #2, picture P_4, DD #3, and picture P_6 in this order.

[0140]

Also in the example in part B of Fig. 16, the first AU is composed of AD #1, picture P_1, DD #1, and picture P_2, and the second AU is composed of AD #2, picture P_3, DD #2, and picture P_4. The third AU is composed of AD #3, picture P_5, DD #3, and picture P_6.

[0141]

An AD represents the start position of data of a

picture in the Base stream and a picture in the Dependent stream, which are obtained at the same time point, that is, represents the start position of image encoding bit streams obtained from all viewpoints at a predetermined time. In addition, a DD represents the boundary position of data of a picture in the Dependent stream, that is, represents the start position of an image encoding bit stream in the Dependent stream.

[0142]

[Regarding Syntax]

The syntax of a DD will be described.

[0143]

All encoded data items including an AD and pictures are stored in a unit called a NAL unit, and are rearranged in a predetermined order to configure a bit stream. The details are described in ISO/IEC 14496-10 | ITU-T H.264.

[0144]

A DD is also stored in a NAL unit.

[0145]

Fig. 17 is a diagram illustrating an example configuration of a NAL unit.

[0146]

Encoded data items and the header are handled byte-by-byte. NAL_BytesInNALunit is a parameter indicating the size of the NAL unit in bytes, which is necessary to identify NAL

unit boundaries.

[0147]

One method for identifying NAL unit boundaries is to search for start codes. In other cases, it is necessary to transmit the NAL unit size to a decoder using some method. rbsp_byte[] represents encoded data items or the header that are stored in a NAL unit.

[0148]

Fig. 18 is a diagram illustrating the syntax of a NAL unit.

[0149]

forbidden_zero_bit is 1-bit data, which is always 0.

[0150]

nal_ref_idc is 2-bit data, which indicates that the data items in the NAL unit are referred to by other NAL units if the nal_ref_idc value is a value other than 0. If the nal_ref_idc value is 0, the data items are not referred to by other NAL units.

[0151]

nal_unit_type is 5-bit data, which indicates the content of the NAL unit.

[0152]

forbidden_zero_bit, nal_ref_idc, and nal_unit_type constitute the header in Fig. 17.

[0153]

Figs. 19 and 20 are diagrams illustrating the definition of nal_unit_type.

[0154]

As illustrated in Fig. 19, nal_unit_type = 9 indicates that the content of the NAL unit is an AD.

[0155]

As illustrated in Fig. 20, nal_unit_type = 18 indicates that the content of the NAL unit is a DD.

[0156]

The simplified syntax of the NAL unit in Fig. 18 is as illustrated in Fig. 21.

[0157]

The NAL unit has the start code added therein in accordance with the syntax illustrated in Fig. 22, and constitutes a stream called a byte stream. The byte stream corresponds to the bit stream described above.

[0158]

startcode_prefix (start_code_prefix_one_3bytes) of 3 bytes (0x000001) in Fig. 22 is a start code to be added to the beginning of the NAL unit. startcode_prefix is a unique code that must not appear at other places in the byte stream.

[0159]

Searching for startcode_prefix in a byte stream results in detecting the beginning of a NAL unit. In addition, checking nal_unit_type at the header of the NAL unit whose

beginning has been detected makes it possible to access a desired NAL unit.

[0160]

A simplification of the syntax in Fig. 22 is as illustrated in Fig. 23.

[0161]

In this manner, DDs are placed at boundaries of pictures in a Dependent stream regardless of whether the Dependent stream is multiplexed with the Base stream into a single bit stream or is included in another bit stream.

[0162]

A start code is added to the beginning of a NAL unit storing a DD, and the nal_unit_type value of the NAL unit is set to 18, which indicates that the content is a DD.

[0163]

The multi-viewpoint decoding apparatus can detect the beginning of a NAL unit by detecting a start code. In addition, checking that the nal_unit_type value of the NAL unit whose beginning has been detected is 18 makes it possible to detect a DD, that is, a picture boundary (beginning) in a Dependent stream.

[0164]

[Multi-Viewpoint Decoding Apparatus]

Fig. 24 is a block diagram illustrating an example configuration of a dependent stream decoding unit that is a

portion of a multi-viewpoint decoding apparatus according to an embodiment of the present invention.

[0165]

A multi-viewpoint decoding apparatus according to an embodiment of the present invention has the same configuration as the configuration in Fig. 8. Fig. 24 illustrates an example configuration of the dependent stream decoding unit 115 in Fig. 8. A video decoding unit 112 of a multi-viewpoint decoding apparatus 101 according to an embodiment of the present invention has the same configuration as the configuration in Fig. 9.

[0166]

The configuration of the dependent stream decoding unit 115 illustrated in Fig. 24 is the same configuration as the configuration in Fig. 10, except for the point that DDs are output from the lossless encoding/decoding unit 142.

[0167]

The lossless encoding/decoding unit 142 decodes a DD included in a Dependent stream, and outputs the DD. Based on the output DD, the start position of decoding or the like at the time of random access is specified by an upper application or the like.

[0168]

An MVC virtual buffer model will be described.

[0169]

Fig. 25 is a diagram illustrating an example of a virtual buffer model in a case where an MVC bit stream is composed of two streams, that is, a Base stream and a Dependent stream.

[0170]

An input Base stream is stored in a buffer 151A, and a Dependent stream is stored in a buffer 151B. Bit streams are read instantaneously from the buffers 151A and 151B on an AU-by-AU basis, and are decoded instantaneously by a video decoding apparatus 152.

[0171]

It is to be noted that the buffer 151A in Fig. 25 corresponds to the buffer 111 in the multi-viewpoint decoding apparatus 101 in Fig. 8 and that the buffer 151B in Fig. 25 corresponds to the buffer 114 in the multi-viewpoint decoding apparatus 101 in Fig. 8. In addition, the video decoding unit 152 in Fig. 25 corresponds to the video decoding unit 112 and the dependent stream decoding unit 115 in the multi-viewpoint decoding apparatus 101 in Fig. 8.

[0172]

For example, in a case where the Base stream in part B of Fig. 16 is stored in the buffer 151A, and the Dependent stream is stored in the buffer 151B, at a certain timing, the picture P_1 is read from the buffer 151A in accordance with the AD #1, and the picture P_2 is read from the buffer

151B in accordance with the DD #1. The read pictures are decoded by the video decoding unit 152.

[0173]

In addition, at the next timing, the picture P_3 is read from the buffer 151A in accordance with the AD #2, and the picture P_4 is read from the buffer 151B in accordance with the DD #2.

[0174]

A virtual buffer model in a case where an MVC bit stream is composed of a single stream having a Base stream and a Dependent stream is the same as that illustrated in Fig. 11.

[0175]

For example, in a case where the bit stream in part A of Fig. 16 is stored in the buffer 151 in Fig. 11, at a certain timing, the pictures P_1 and P_2 are read from the buffer 151 in accordance with the AD #1.

[0176]

In addition, at the next timing, the pictures P_3 and P_4 are read from the buffer 151 in accordance with the AD #2.

[0177]

Next, an example of a method for accessing the beginning of a predetermined AU at the time of random access, at the time of occurrence of an error, or the like will be described with reference to Fig. 26.

[0178]

An input bit stream is first supplied to a buffer 211 and is temporarily stored. Thereafter, the bit stream is supplied to a start code detection unit 212 and a decoder 214.

[0179]

The start code detection unit 212 detects a start code having a unique data string. Upon detection of a start code, the start code detection unit 212 outputs subsequent data to a syntax analysis unit 213.

[0180]

The syntax analysis unit 213 analyzes the data supplied from the start code detection unit 212. Because of the data subsequent to the start code, the data supplied from the start code detection unit 212 is data of a NAL unit.

[0181]

Upon identification of the start position of an AU by analyzing the header or the like of a NAL unit, the syntax analysis unit 213 outputs a signal indicating an AU start position to the decoder 214 and the buffer 211.

[0182]

In a case where an AD is present (in a case where data of a NAL unit is supplied), the syntax analysis unit 213 checks whether or not nal_unit_type = 9 is satisfied. In a case where nal_unit_type = 9 is satisfied, as described

above, the content of the NAL unit is an AD. Thus, the beginning of the AU is detected.

[0183]

Similarly, in a case where an AD is present, the syntax analysis unit 213 checks whether or not nal_unit_type = 18 is satisfied. In a case where nal_unit_type = 18 is satisfied, as described above, the content of the NAL unit is a DD. Thus, the beginning of a picture in the Dependent stream is detected.

[0184]

The decoder 214 and the buffer 211 subsequently perform decoding of pictures on an AU-by-AU basis in a normal manner in accordance with the signal from the syntax analysis unit 213.

[0185]

The operation of detecting the beginning of a picture in a Dependent stream will be described with reference to a flowchart in Fig. 27.

[0186]

Here, as illustrated in part B of Fig. 16, it is assumed that a Base stream and a Dependent stream constitute different bit streams and that the Dependent stream has an encoded DD.

[0187]

In step S1, the start code detection unit 212 searches

for a start code.

[0188]

After a start code is detected, in step S2, the syntax analysis unit 213 checks whether or not nal_unit_type = 18 is satisfied.

[0189]

In a case where it is determined in step S2 that nal_unit_type = 18 is satisfied, in step S3, the syntax analysis unit 213 detects that the NAL unit having the detected start code is a unit storing a DD and is at the beginning of a picture in the Dependent stream.

[0190]

The operation in a case where, as illustrated in part B of Fig. 14, the stream is composed of two bit storms and where, as in conventional MVC, the Dependent stream has no DDs is illustrated in Fig. 28.

[0191]

As illustrated in Fig. 28, in a case where the Dependent stream has no DDs, slice headers are also decoded, and multiple conditional determinations are performed. Finally, in step S23, it is possible to detect the beginning of an AU (beginning of a picture).

[0192]

In Fig. 28, the values used in the respective determinations up to step S23 are pieces of information

written in slice headers stored in RBSPs (Fig. 17) of a NAL unit. Unlike nal_unit_type = 18 described above, the above pieces of information are not written in headers of NAL units, and therefore, complicated analysis processing is performed.

[0193]

[Three-Viewpoint Bit Stream]

Three-viewpoint bit streams will be described.

[0194]

Fig. 29 is a diagram illustrating the configuration of AUs obtained by encoding three-viewpoint video using MVC.

[0195]

As described above, in the case of MVC, one AU stores the data of all views obtained at the same time point. The same applies even in a case where the number of viewpoints is three.

[0196]

In Fig. 29, AU #11 is composed of picture P_{11} in a Base stream, picture P_{12} in a Dependent stream 1, and picture P_{13} in a Dependent stream 2.

[0197]

AU #12 is composed of picture P_{14} in the Base stream, picture P_{15} in the Dependent stream 1, and picture P_{16} in the Dependent stream 2.

[0198]

AU #13 is composed of picture P_{17} in the Base stream, picture P_{18} in the Dependent stream 1, and picture P_{19} in the Dependent stream 2.

[0199]

Fig. 30 includes diagrams illustrating an example of the structure of bit streams.

[0200]

The bit streams illustrated in Fig. 30 include no DDs.

[0201]

Part A of Fig. 30 illustrates an example of a structure in a case where the Base stream, the Dependent stream 1, and the Dependent stream 2 are multiplexed into a single bit stream.

[0202]

Part B of Fig. 30 illustrates an example of a structure in a case where the Base stream, the Dependent stream 1, and the Dependent stream 2 are included in a total of three different bit streams. Portions corresponding to the portions illustrated in part A of Fig. 30 are assigned the same symbols.

[0203]

For example, in a case where the bit stream in part A of Fig. 30 is supplied, after detecting the AD #11, the video decoding unit 152 sequentially reads the picture P_{11} in the Base stream, the picture P_{12} in the Dependent stream 1,

and the picture P_{13} in the Dependent stream 2, and individually decodes the pictures.

[0204]

In addition, in a case where the bit streams in part B of Fig. 30 are supplied, for the first AU in the Base stream, the video decoding unit 152 detects the AD #11 and reads and decodes the picture P_{11}. In addition, the video decoding unit 152 detects, for the second AU, the AD #12 and reads and decodes the picture P_{14}.

[0205]

The Dependent streams 1 and 2 have no DDs. Therefore, in order to read the individual pictures, the need exists to analyze the syntax in the bit streams in accordance with the process in Fig. 28 and to detect the boundaries of the pictures P_{12}, P_{13}, P_{15}, P_{16}, P_{18}, and P_{19}.

[0206]

Fig. 31 includes diagrams illustrating an example of the structure of bit streams generated by a multi-viewpoint encoding apparatus 101 according to an embodiment of the present invention.

[0207]

Part A of Fig. 31 illustrates an example of a structure in a case where the Base stream, the Dependent stream 1, and the Dependent stream 2 are multiplexed into a single bit stream.

[0208]

Part B of Fig. 31 illustrates an example of a structure in a case where the Base stream, the Dependent stream 1, and the Dependent stream 2 are included in a total of three different bit streams.

[0209]

In a case where the bit stream in part A of Fig. 31 is supplied, after detecting the AD #11, the video decoding unit 152 sequentially reads the picture P_{11} in the Base stream, the picture P_{12} in the Dependent stream 1, and the picture P_{13} in the Dependent stream 2, and individually decodes the pictures.

[0210]

In a case where the bit streams in part B of Fig. 31 are supplied, for the first AU in the Base stream, after detecting the AD #11, the video decoding unit 152 decodes the picture P_{11}. For the second AU, after detecting the AD #12, the video decoding unit 152 decodes the picture P_{14}.

[0211]

In addition, for the Dependent stream 1, after detecting the DD #11, the video decoding unit 152 decodes the picture P_{12}. In addition, after detecting the DD #12, the video decoding unit 152 decodes the picture P_{15}.

[0212]

Further, for the Dependent stream 2, after detecting

the DD #21, the video decoding unit 152 decodes the picture P_{13}. In addition, after detecting the DD #22, the video decoding unit 152 decodes the picture P_{16}.

[0213]

In this manner, in the multi-viewpoint decoding apparatus 101, an operation is performed so that a desired AU is detected and decoded in accordance with an AD that is a piece of information illustrating an AU boundary and a DD that is a piece of information illustrating a boundary between pictures in a Dependent stream and so that a stereoscopic image is displayed.

[0214]

Since there is no need to analyze slice headers in the case of detecting the start position of a certain picture in a Dependent stream, high-speed random access is made feasible.

[0215]

The series of processes described above can be executed by hardware or software. In a case where the series of processes is executed by software, a program constituting the software is installed from a program recording medium into a computer incorporated in dedicated hardware, a general-purpose personal computer, or the like.

[0216]

Fig. 32 is a block diagram illustrating an example

configuration of hardware of a computer that executes the series of processes described above in accordance with a program.

[0217]

A CPU (Central Processing Unit) 301, a ROM (Read Only Memory) 302, and a RAM (Random Access Memory) 303 are connected to one another via a bus 304.

[0218]

An input/output interface 305 is further connected to the bus 304. An input unit 306 having a keyboard, a mouse, and the like and an output unit 307 having a display, speakers, and the like are connected to the input/output interface 305. In addition, a storage unit 308 having a hard disk, a non-volatile memory, and the like, a communication unit 309 having a network interface and the like, and a drive 310 that drives a removable medium 311 are connected to the bus 304.

[0219]

In the computer configured as above, for example, the CPU 301 loads a program stored in the storage unit 308 into the RAM 303 via the input/output interface 305 and the bus 304, and executes the program. Thus, the series of processes described above is performed.

[0220]

The program executed by the CPU 301 is recorded on, for

example, the removable medium 311, or is provided via a wired or wireless transmission medium such as a local area network, the Internet, or digital broadcasting and is installed into the storage unit 308.

[0221]

It is to be noted that the program executed by the computer may be a program in accordance with which the processes are performed in a time-series manner in the order described herein, or may be a program in accordance with which the processes are performed in parallel or at a required timing such as when called.

[0222]

Embodiments of the present invention are not to be limited to the embodiment described above, and a variety of changes can be made without departing from the scope of the present invention.

Reference Signs List

[0223]

41 multi-viewpoint encoding apparatus, 51 rearrangement buffer, 52 video encoding unit, 53 frame memory, 54 rearrangement buffer, 55 dependent stream encoding unit, 56 frame memory, 57 multiplexing unit, 101 multi-viewpoint decoding apparatus, 102 3D display, 111 buffer, 112 video decoding unit, 113 frame memory, 114 buffer, 115 dependent stream decoding unit, 116 frame memory, 201 DD encoding unit

CLAIMS

[Claim 1]

An image signal decoding apparatus that decodes a bit stream produced by encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, comprising:

decoding means for decoding a first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint, identifying the start of encoded data obtained at the predetermined time point,

decoding a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint, identifying the start of encoded data of an image signal obtained from the other viewpoint, and decoding the bit stream.

[Claim 2]

The image signal decoding apparatus according to Claim 1, wherein the image signals obtained from the plurality of viewpoints are image signals obtained from two viewpoints, and are configured as a single bit stream.

[Claim 3]

The image signal decoding apparatus according to Claim 1, wherein image signals obtained from a plurality of viewpoints are image signals obtained from two viewpoints, and are configured as two bit streams.

[Claim 4]

The image signal decoding apparatus according to Claim 3, wherein the decoding means decodes the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at the predetermined time point, the first unique data string being encoded in an image obtained from one of the viewpoints, identifies the start of encoded data obtained at the predetermined time point,

decodes the second unique data string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point, the second unique data string being encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint, identifies the start of encoded data of an image signal obtained from the other viewpoint, and decodes the bit stream.

[Claim 5]

The image signal decoding apparatus according to Claim 1, wherein in the case of accessing a position at which

decoding can be started correctly,

the decoding means decodes the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at the predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint, identifies the start of encoded data obtained at the predetermined time point from the one viewpoint,

decodes the second unique data strings indicating the start of the image encoded bit stream obtained from the other viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of the encoded bit stream of the image obtained from the other viewpoint, identifies the start of the encoded data of the image signal obtained from the other viewpoint, and decodes the bit stream from identified start positions.

[Claim 6]

An image signal decoding method for decoding a bit stream produced by encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, comprising the steps of:

decoding a first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, the first unique

data string being encoded in an image obtained from one viewpoint; identifying the start of encoded data obtained at the predetermined time point;

decoding a second unique data string indicating the start of an image encoded bit stream obtained from another viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint; identifying the start of encoded data of an image signal obtained from the other viewpoint; and decoding the bit stream.

[Claim 7]

A program for causing a computer to execute a process for decoding a bit stream produced by encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, the program causing the computer to execute a process comprising the steps of:

decoding a first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, the first unique data string being encoded in an image obtained from one viewpoint; identifying the start of encoded data obtained at the predetermined time point;

decoding a second unique data string indicating the start of an image encoded bit stream obtained from another

viewpoint at the predetermined time point, the second unique data string being individually encoded at the beginning of an encoded bit stream of an image obtained from the other viewpoint; identifying the start of encoded data of an image signal obtained from the other viewpoint; and decoding the bit stream.

[Claim 8]

An image signal encoding apparatus that encodes a stereoscopic image signal including image signals obtained from a plurality of viewpoints, comprising:

encoding means for encoding a first unique data string in an image obtained from one viewpoint, the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point, and

individually encoding a second unique data string at the beginning of an encoded bit stream of an image obtained from another viewpoint, the second unique data string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point.

[Claim 9]

The image signal encoding apparatus according to Claim 8, wherein the encoding means multiplexes encoded data obtained from all the viewpoints into a single bit stream.

[Claim 10]

The image signal encoding apparatus according to Claim 8, wherein the encoding means causes encoded data obtained from all the viewpoints to be included in two bit streams.

[Claim 11]

The image signal encoding apparatus according to Claim 8, wherein the image signals obtained from the plurality of viewpoints are image signals obtained from two viewpoints.

[Claim 12]

An image signal encoding method for encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, comprising the steps of:

encoding a first unique data string in an image obtained from one viewpoint, the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point; and

individually encoding a second unique data string at the beginning of an encoded bit stream of an image obtained from another viewpoint, the second unique data string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point.

[Claim 13]

A program for causing a computer to execute a process for encoding a stereoscopic image signal including image signals obtained from a plurality of viewpoints, the program causing the computer to execute a process comprising the

steps of:

encoding a first unique data string in an image obtained from one viewpoint, the first unique data string indicating the start of image encoded bit streams obtained from all the viewpoints at a predetermined time point; and

individually encoding a second unique data string at the beginning of an encoded bit stream of an image obtained from another viewpoint, the second unique data string indicating the start of an image encoded bit stream obtained from the other viewpoint at the predetermined time point.