DESCRIPTION

IMAGE PROCESSING DEVICE AND IMAGE PROCESSING METHOD

TECHNICAL FIELD

[0001]

The present disclosure relates to an image processing device and an image processing method. Specifically, the present disclosure relates to an image processing device and an image processing method capable of improving encoding efficiency.

BACKGROUND ART

[0002]

In recent years, devices, which handle image information as digital data, which, in such a case, aim to transmit and store information with a high efficiency, and which conform to a scheme such as MPEG (Moving Picture Experts Group), for compressing image information using orthogonal transformation, such as discrete cosine transformation, and using motion compensation by utilizing redundancy that is unique to the image information have become widespread in both information distribution in broadcasting stations and information reception in ordinary homes.

[0003]

In particular, MPEG2 (International Organization for Standardization and International Electrotechnical Commission (ISO/IEC) 13818-2) is defined as a general-purpose image encoding scheme and is presently widely used in a wide range of applications for professional use and consumer use as standards for both interlaced-

scanning images and sequential scanning images and standard and high-definition images. By employing the MPEG2 compression scheme, for example, a coding rate (bit rate) of 4 to 8 Mbps is allocated for an interlaced-scanning image of a standard resolution with 720 × 480 pixels, and a coding rate (bit rate) of 18 to 22 Mbps is allocated for an interlaced-scanning image of a high resolution with 1920 × 1088 pixels. As a result, a high compression ratio and a good image quality can be realized.

[0004]

The MPEG2 has been mainly intended for high-image-quality encoding appropriate for broadcasting, but was not compatible with an encoding scheme for realizing a lower coding rate (bit rate) (a higher compression ratio) than that of MPEG1. With the popularity of mobile terminals, the demand for such an encoding scheme is expected to increase in the future. To respond this, standardization of MPEG4 encoding schemes have been confirmed. With regard to an image encoding scheme, the specification thereof was confirmed as the international standard ISO/IEC 14496-2 in December in 1998.

[0005]

In addition, in recent years, originally for the purpose of video coding for television conferencing, standardization of specifications of a standard called H.26L (ITU-T (International Telecommunication Union Telecommunication Standardization Sector) Q6/16 VCEG (Video Coding Expert Group)) has progressed. H.26L is known to achieve higher encoding efficiency although it requires a greater amount of computations for encoding

and decoding than conventional encoding schemes such as MPEG2 and MPEG4. Moreover, currently, as part of the activity of MPEG4, standardization for incorporating functions, which are not supported by H.26L, into the H.26L is performed as Joint Model of Enhanced-Compression Video Coding to realize high encoding efficiency.

[0006]

The schedule of standardization showed that, it became an international standard under the name of H.264 and MPEG-4 Part 10 (Advanced Video Coding, hereinafter referred to as AVC) in March, 2003.

[0007]

However, setting the size of a macroblock to 16 × 16 pixels is not optimal for a large image frame named UHD (Ultra High Definition; 4000 × 2000 pixels) that will become an object of the next generation encoding scheme.

[0008]

Thus, the standardization of an encoding system called HEVC (High Efficiency Video Coding) has been currently developed by JCTVC (Joint Collaboration Team-Video Coding), which is a joint standardization organization of ITU-T and ISO/IEC, for the purpose of further improving the encoding efficiency compared to AVC (for example, see Non-Patent Document 1).

[0009]

In the HEVC encoding scheme, a coding unit (CU) is defined as the same processing unit as the macroblock in the AVC scheme. The size of CU is not fixed to 16 × 16 pixels unlike the macroblock of the AVC scheme but is designated in image compression information in respective sequences.

[0010]

However, in order to improve encoding of motion vectors using median prediction defined in the AVC scheme, a method that allows "temporal predictor" and "spatio-temporal predictor" as well as "spatial predictor" to be used as candidates for predictive motion vectors has been taken into consideration (for example, see Non-Patent Document 2).

[0011]

Moreover, a method called motion partition merging in which merge_flag and merge_left_flag are transmitted is proposed as one of encoding schemes for motion information (for example, see Non-Patent Document 3).

CITATION LIST

NON-PATENT DOCUMENT

[0012]

Non-Patent Document 1: Thomas Wiegand, Woo-Jin Han, Benjamin Bross, Jens-Rainer Ohm, Gary J. Sullivan, "Working Draft 1 of High-Efficiency Video Coding", JCTVC-C403, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 3rd Meeting: Guangzhou, CN, 7-15 October, 2010

Non-Patent Document 2: Joel Jung, Guillaume Laroche, "Competition-Based Scheme for Motion Vector Selection and Coding," VCEG-AC06, ITU-Telecommunications Standardization Sector STUDY GROUP 16 Question 6, Video Coding Experts Group (VCEG) 29th Meeting: Klagenfurt, Austria, 17-18 July, 2006

Non-Patent Document 3: Martin Winken, Sebastian Bosse, Benjamin Bross, Philipp Helle, Tobias Hinz, Heiner

Kirchhoffer, Haricharan Lakshman, Detlev Marpe, Simon Oudin, Matthias Preiss, Heiko Schwarz, Mischa Siekmann, Karsten Suehring, and Thomas Wiegand, "Description of video coding technology proposed by Fraunhofer HHI," JCTVC-A116, April, 2010

SUMMARY OF THE INVENTION

PROBLEMS TO BE SOLVED BY THE INVENTION

[0013]

However, these techniques illustrate processes in the same view only and cannot perform inter-view vector prediction in the case of multi-view encoding. Thus, encoding efficiency may decrease.

[0014]

Embodiments of the present invention are made in view of the situations, and aim to suppress a decrease in the encoding efficiency.

SOLUTIONS TO PROBLEMS

[0014a]

According to a first aspect of the present invention, there is provided an image processing device including: circuitry configured to set a candidate of a predictive vector by setting a co-located vector as unavailable as the candidate of the predictive vector for a coding vector in a current block of a current picture where a reference picture of the coding vector is of a different type of two reference picture types than a reference picture of the co-located vector, wherein the co-located vector is in a co-located block included in a co-located picture of a time different from the current picture but of a same view as the current picture, and wherein the two reference picture types comprise a long-term reference picture type and a short-term reference picture type; reconstruct the coding vector using the predictive vector in the current block according to the candidate of the predictive vector and a difference vector between the coding vector and the generated predictive vector; and decode the

current block using the coding vector.

[0014b]

According to a second aspect of the present invention, there is provided an image processing method, including: setting a candidate of a predictive vector by setting a co-located vector as unavailable as the candidate of the predictive vector for a coding vector in a current block of a current picture where a reference picture of the coding vector is of a different type of two reference picture types than a reference picture of the co-located vector, wherein the co-located vector is in a co-located block included in a co-located picture of a time different from the current picture but of a same view as the current picture, and wherein the two reference picture types comprise a long-term reference picture type and a short-term reference picture type; reconstructing the coding vector using the predictive vector in the current block according to the candidate of the predictive vector and a difference vector between the coding vector and the generated predictive vector; and decoding the current block using the coding vector.

[0015]

Disclosed herein is an image processing device including: a predictive vector generating unit that generates a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when encoding the current parallax vector; and a difference vector generating unit that generates a difference vector between the current parallax vector and the predictive vector generated by the predictive vector generating unit.

[0016]

The predictive vector generating unit may generate a predictive vector of the current parallax vector using a parallax vector of a co-located block included in a co-located

picture of a time different from a current picture of the same view as a current view.

[0017]

The predictive vector generating unit may set the co-located block to be available when a property of a vector of the current block is identical to a property of a vector of the co-located block.

[0018]

The property of the vector is a type of a vector, and the predictive vector generating unit may set the co-located block to be available when the property of the vector of the current block is a parallax vector and the property of the vector of the co-located block is a parallax vector.

[0019]

The predictive motion vector generating unit may determine the property of the vector of the current block and the property of the vector of the co-located block using POC (Picture Order Count) that indicates an output order of pictures.

[0020]

The predictive motion vector generating unit may determine the property of the vector of the current block and the property of the vector of the co-located block using the POC of the current picture, the POC of a current reference picture referred from the current picture, the POC of the co-located picture, and the POC of a co-located reference picture referred from the co-located picture.

[0021]

The predictive motion vector generating unit may determine that the property of the vector of the current block and the property of the vector of the co-located block are parallax vectors when the POC of the current picture is identical to the

POC of the current reference picture referred from the current picture and the POC of the co-located picture is identical to the POC of the co-located reference picture referred from the co-located picture.

[0022]

The predictive vector generating unit may set the co-located block to be not available when the property of the vector of the current block is different from the property of the vector of the co-located block.

[0023]

The property of the vector is a type of a reference picture, and the predictive vector generating unit may set the co-located block to be not available when the type of the reference picture of the current block is different from the type of the reference picture of the co-located block.

[0024]

The property of the vector is a type of a reference picture, and the predictive vector generating unit may skip a process of searching a reference index when the type of the reference picture of the current block is a long reference type and the type of the reference picture of the co-located block is a long reference type.

[0025]

The predictive vector generating unit may generate a predictive vector of the current parallax vector using a parallax vector of a reference block included in a picture of the same time as a current picture of a view different from the current view.

[0026]

The predictive vector generating unit may scale the reference parallax vector based on a positional relationship between a current picture and a reference picture referred when

generating a predictive motion vector to generate a predictive vector of the current parallax vector.

[0027]

The predictive vector generating unit may generate a predictive vector of the current motion vector using a reference motion vector referred when generating a predictive motion vector, when encoding the current motion vector of the current block used in prediction using correlation in a temporal direction, and the difference vector generating unit may generate a difference vector between the current motion vector and the predictive vector generated by the predictive vector generating unit.

[0028]

The predictive vector generating unit may generate the predictive vector of the current motion vector using a motion vector of the reference block included in a picture of the same time as the current picture of a view different from the current view.

[0029]

The predictive vector generating unit may generate a predictive vector of the current motion vector using a motion vector of a reference block included in a picture of a time different from the current picture of the same view as the current view.

[0030]

The predictive vector generating unit may scale the reference motion vector based on a positional relationship between the current picture and a reference picture referred when generating a predictive motion vector to generate a predictive vector of the current motion vector.

[0031]

The predictive vector generating unit may generate the

predictive vector using a vector of a block located at the same position as the current block in a state where a position of a pixel of a picture of the same time as the current picture of a view different from the current view is shifted.

[0032]

The predictive vector generating unit may set a shift amount of the image according to a parallax vector of a neighboring region of the current block.

[0033]

The predictive vector generating unit may use a parallax vector in an X-direction, of the neighboring block in which the value of a parallax vector in a Y-direction is not zero as the shift amount.

[0034]

The predictive vector generating unit may use a value calculated from parallax vectors in an X-direction, of a plurality of the neighboring blocks in which the value of a parallax vector in a Y-direction is not zero as the shift amount.

[0035]

The predictive vector generating unit may use an average value or a median value of the parallax vectors in the X-direction, of the plurality of the neighboring blocks in which the value of the parallax vector in the Y-direction is not zero as the shift amount of the image.

[0036]

The predictive vector generating unit may set the shift amount of the image according to a global parallax vector.

[0037]

Disclosed herein is an image processing method of an image processing device, for allowing the image processing device to execute: generating a predictive vector of a current parallax

vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when encoding the current parallax vector; and generating a difference vector between the current parallax vector and the generated predictive vector.

[0038]

Disclosed herein is an image processing device including: a predictive vector generating unit that generates a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when decoding the current parallax vector; and an arithmetic unit that performs an operation of adding the predictive vector generated by the predictive vector generating unit to a difference vector between the current parallax vector and the predictive vector to reconstruct the current parallax vector.

[0039]

Disclosed herein is an image processing method of an image processing device, for allowing the image processing device to execute: generating a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when decoding the current parallax vector; and performing an operation of adding the generated predictive vector to a difference vector between the current parallax vector and the predictive vector to reconstruct the current parallax vector.

[0040]

Disclosed herein is an image processing device including: a predictive vector generating unit that sets a co-located

block to be not available when a type of a reference picture of a current block is different from a type of the reference picture of a co-located block included in a co-located picture of a different time from a current picture when encoding a current motion vector of the current block used in prediction using correlation in a temporal direction and generates a predictive vector of the current motion vector using a reference motion vector referred when generating a predictive motion vector; and a difference vector generating unit that generates a difference vector between the current motion vector and the predictive vector generated by the predictive vector generating unit.

[0041]

Disclosed herein is an image processing method of an image processing device, for allowing the image processing device to execute: setting a co-located block to be not available when a type of a reference picture of a current block is different from a type of the reference picture of a co-located block included in a co-located picture of a different time from a current picture when encoding a current motion vector of the current block used in prediction using correlation in a temporal direction, and generating a predictive vector of the current motion vector using a reference motion vector referred when generating a predictive motion vector; and generating a difference vector between the current motion vector and the generated predictive vector.

[0042]

In one embodiment of the present invention, a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction is generated using a reference parallax vector referred when generating a predictive motion vector, when encoding the

current parallax vector; and a difference vector between the current parallax vector and the generated predictive vector is generated.

[0043]

According to another embodiment of the present invention, a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction is generated using a reference parallax vector referred when generating a predictive motion vector, when decoding the current parallax vector; and an operation of adding the generated predictive vector to a difference vector between the current parallax vector and the predictive vector to reconstruct the current parallax vector is performed.

[0044]

According to a still another embodiment of the present invention, a co-located block is set to be not available when a type of a reference picture of a current block is different from a type of the reference picture of a co-located block included in a co-located picture of a different time from a current picture when encoding a current motion vector of the current block used in prediction using correlation in a temporal direction; a predictive vector of the current motion vector is generated using a reference motion vector referred when generating a predictive motion vector; and a difference vector between the current motion vector and the generated predictive vector is generated.

EFFECTS OF THE INVENTION

[0045]

According to embodiments of the present invention, it is possible

to process images. In particular, it is possible to suppress a decrease in encoding efficiency.

BRIEF DESCRIPTION OF THE DRAWINGS

[0046]

Fig. 1 is a diagram for describing an example of the types of prediction units.

Fig. 2 is a diagram for describing an example of neighboring regions used for reference image index determination in a merge mode temporal correlation region.

Fig. 3 is a diagram for describing an example of reference image index determination conditions in a merge mode temporal correlation region.

Fig. 4 is a diagram for describing an example of a temporal correlation region determination method.

Fig. 5 is a diagram for describing an example of a method of determining an image that includes a temporal correlation region.

Fig. 6 is a diagram for describing an example of positional relationship between a current region and a temporal correlation region.

Fig. 7 is a diagram illustrating an example of scaling of a motion vector of a temporal correlation region.

Fig. 8 is a diagram illustrating an example of a reference relationship of a 3-view image.

Fig. 9 is a diagram for describing an example of allocation of a reference image when predicting a parallax vector with respect to a current region.

Fig. 10 is a diagram for describing an example of allocation of a reference image when predicting a motion

vector with respect to a current region.

Fig. 11 is a block diagram illustrating a main configuration example of an image encoding device.

Fig. 12 is a block diagram illustrating a main configuration example of a motion parallax prediction/compensation unit.

Fig. 13 is a block diagram illustrating a main configuration example of a temporal parallax correlation predictive vector generating unit.

Fig. 14 is a diagram for describing an example of scheme selection.

Fig. 15 is a diagram for describing the way of determining availability of a co-located vector.

Fig. 16 is a diagram for describing an example in which a co-located vector is available.

Fig. 17 is a diagram illustrating an example of syntax of a sequence parameter set.

Fig. 18 is a diagram illustrating an example of syntax of a slice header.

Fig. 19 is a diagram illustrating an example of syntax of a prediction unit.

Fig. 20 is a flowchart for describing an example of the flow of an encoding process.

Fig. 21 is a flowchart for describing an example of the flowchart of an inter-motion prediction process.

Fig. 22 is a flowchart for describing an example of the flow of a merge mode process.

Fig. 23 is a flowchart for describing an example of the flow of a parallax motion vector prediction process.

Fig. 24 is a flowchart for describing an example of the flow of a temporal parallax correlation prediction

process.

Fig. 25 is a flowchart continued from Fig. 22, for describing an example of the flow of the temporal parallax correlation prediction process.

Fig. 26 is a flowchart continued from Fig. 23, for describing an example of the flow of the temporal parallax correlation prediction process.

Fig. 27 is a flowchart continued from Fig. 24, for describing an example of the flow of the temporal parallax correlation prediction process.

Fig. 28 is a flowchart for describing an example of the flow of a scheme-1 process.

Fig. 29 is a flowchart for describing an example of the flow of a scheme-3 process.

Fig. 30 is a flowchart for describing an example of the flow of a scheme-4-2 process.

Fig. 31 is a block diagram illustrating a main configuration example of an image decoding device.

Fig. 32 is a block diagram illustrating a main configuration example of a motion parallax compensation unit.

Fig. 33 is a flowchart for describing an example of the flow of a decoding process.

Fig. 34 is a flowchart for describing an example of the flow of a prediction process.

Fig. 35 is a flowchart for describing an example of the flow of a motion parallax compensation process.

Fig. 36 is a flowchart for describing an example of the flow of a motion parallax vector generation process.

Fig. 37 is a block diagram illustrating another configuration example of the image encoding device.

Fig. 38 is a block diagram illustrating a main configuration example of a motion prediction/compensation unit.

Fig. 39 is a block diagram illustrating a main configuration example of a vector predicting unit.

Fig. 40 is a block diagram illustrating a main configuration example of a different picture-based predictive vector generating unit.

Fig. 41 is a flowchart for describing an example of the flow of a motion prediction/compensation process.

Fig. 42 is a flowchart for describing an example of the flow of a vector prediction process.

Fig. 43 is a flowchart for describing an example of the flow of a predictive vector generation process.

Fig. 44 is a flowchart for describing an example of the flow of a different picture-based predictive vector generation process.

Fig. 45 is a flowchart for describing an example of the flow of a shift amount determining process.

Fig. 46 is a diagram illustrating an example of an arrangement of a current block and neighboring blocks.

Fig. 47 is a block diagram illustrating another configuration example of an image decoding device.

Fig. 48 is a block diagram illustrating a main configuration example of a motion compensation unit.

Fig. 49 is a block diagram illustrating a main configuration example of a vector decoding unit.

Fig. 50 is a block diagram illustrating a main configuration example of a different picture-based predictive vector generating unit.

Fig. 51 is a flowchart for describing an example of

the flow of a motion compensation process.

Fig. 52 is a flowchart for describing an example of the flow of a vector decoding process.

Fig. 53 is a flowchart for describing an example of the flow of a predictive vector generation process.

Fig. 54 is a flowchart for describing an example of the flow of a different picture-based predictive vector generation process.

Fig. 55 is a flowchart for describing an example of the flow of a shift amount determining process.

Fig. 56 is a diagram illustrating an example of the way of generating a predictive vector.

Fig. 57 is a diagram for describing a parallax and a depth.

Fig. 58 is a diagram for describing an example of a predictive vector generating method.

Fig. 59 is a flowchart for describing an example of the flow of a predictive vector generation process.

Fig. 60 is a flowchart for describing an example of the flow of a different picture-based predictive vector generation process.

Fig. 61 is a flowchart continued from Fig. 60, for describing an example of the flow of a different picture-based predictive vector generation process.

Fig. 62 is a diagram for describing an example of the aspect of a reference image for a fixed background application.

Fig. 63 is a diagram for describing an example of the aspect of a reference image for a stereo application.

Fig. 64 is a diagram for comparing examples of reference image types and vector properties.

Fig. 65 is a diagram for describing an example of a neighboring block.

Fig. 66 is a diagram for describing an example of handling of a temporal correlation block and a neighboring block.

Fig. 67 is a flowchart for describing an example of the flow of a PU motion (parallax) vector and reference index generation process.

Fig. 68 is a flowchart for describing an example of the flow of a merge (skip) mode process.

Fig. 69 is a flowchart for describing an example of the flow of a process of generating candidate motion (parallax) vectors from a temporal correlation block.

Fig. 70 is a flowchart for describing an example of the flow of a process of determining the presence of a scaling process for a motion (parallax) vector of a temporal correlation block and the presence of a candidate.

Fig. 71 is a flowchart for describing an example of the flow of an AMVP mode process.

Fig. 72 is a flowchart for describing an example of the flow of a process of generating candidate motion (parallax) vectors from spatially neighboring blocks.

Fig. 73 is a flowchart for describing an example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the left side.

Fig. 74 is a flowchart continued from Fig. 73, for describing an example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the left side.

Fig. 75 is a flowchart for describing an example of

the flow of a process of determining the presence of a scaling process for a motion (parallax) vector of a neighboring block and the presence of a candidate.

Fig. 76 is a flowchart for describing an example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 77 is a flowchart continued from Fig. 76, for describing an example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 78 is a flowchart for describing an example of the flow of a PU motion (parallax) vector and reference index generation process.

Fig. 79 is a flowchart for describing an example of the flow of a merge (skip) mode process.

Fig. 80 is a flowchart for describing an example of the flow of a process of generating candidate motion (parallax) vectors from a temporal correlation block.

Fig. 81 is a flowchart for describing an example of the flow of an AMVP mode process.

Fig. 82 is a diagram for describing an example of handling of a neighboring block.

Fig. 83 is a flowchart for describing another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the left side.

Fig. 84 is a flowchart continued from Fig. 83, for describing another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the left side.

Fig. 85 is a flowchart for describing an example of the flow of a process of determining the presence of a

candidate motion (parallax) vector for a neighboring block.

Fig. 86 is a flowchart for describing another example of the flow of a process of determining the presence of a scaling process for a motion (parallax) vector of a neighboring block and the presence of a candidate.

Fig. 87 is a flowchart for describing another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 88 is a flowchart continued from Fig. 87, for describing another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 89 is a flowchart continued from Fig. 88, for describing another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 90 is a diagram for describing still another example of handling of a temporal correlation block and a neighboring block.

Fig. 91 is a flowchart for describing still another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the left side.

Fig. 92 is a flowchart continued from Fig. 91, for describing still another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the left side.

Fig. 93 is a flowchart for describing still another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 94 is a flowchart continued from Fig. 93, for describing still another example of the flow of a process of generating candidate motion (parallax) vectors from blocks on the upper side.

Fig. 95 is a block diagram illustrating a main configuration example of a personal computer.

Fig. 96 is a block diagram illustrating an example of a schematic configuration of a television apparatus.

Fig. 97 is a block diagram illustrating an example of a schematic configuration of a mobile phone.

Fig. 98 is a block diagram illustrating an example of a schematic configuration of a recording/reproducing apparatus.

Fig. 99 is a block diagram illustrating an example of a schematic configuration of an imaging apparatus.

MODE FOR CARRYING OUT THE INVENTION

[0047]

Hereinafter, modes (hereinafter referred to as embodiments) for carrying out the present disclosure will be described. The description will be given in the following order:

1. First embodiment (Image encoding device)
2. Second embodiment (Image decoding device)
3. Third embodiment (Image encoding device)
4. Fourth embodiment (Image decoding device)
5. Fifth embodiment (Image encoding device and Image decoding device)
6. Sixth embodiment (Image encoding device and Image decoding device)
7. Seventh embodiment (Computer)

8. Eighth embodiment (Application example)

[0048]

[1. First embodiment]

[Motion prediction]

In image encoding such as AVC (Advanced Video Coding) or HEVC (High Efficiency Video Coding), motion prediction using correlation in the temporal direction (between frames) is performed.

[0049]

The AVC defines layered blocks such as macroblocks or sub-macroblocks as processing units of such a prediction process, and the HEVC defines coding units (CUs).

[0050]

The CU which is also called a coding tree block (CTB) is a partial region of a picture-base image, which plays the same role as the macroblock in the AVC. The size of the macroblock is fixed to 16 × 16 pixels whereas the size of the CU is not fixed but is designated in image compression information in respective sequences.

[0051]

For example, the largest size (LCU: Largest Coding Unit) and the smallest size (SCU: Smallest Coding Unit) of the CU are defined in a sequence parameter set (SPS) included in output encoded data.

[0052]

Each LCU can be split into CUs of the smaller size that is not smaller than the size of SCU by setting split_flag = 1. A CU having the size of 2N × 2N is split into CUs having the size of N × N which is one layer below when the value of split_flag is "1."

[0053]

Further, the CU is split into prediction units (PUs) which are regions (partial regions of a picture-base image) serving as processing units of intra or inter-prediction. Moreover, the CU is split into transform units (TUs) which are regions (partial regions of a picture-base image) serving as processing units of orthogonal transform. Presently, the HEVC can use 16 × 16 and 32 × 32 orthogonal transform in addition to 4 × 4 and 8 × 8 orthogonal transform.

[0054]

In encoding schemes in which CUs are defined and various processes are performed in units of CUs as in the HEVC, it can be regarded that macroblocks in the AVC correspond to LCUs. However, the CUs have a layer structure, the size of LCUs on the uppermost layer is generally set to 128 × 128 pixels, for example, that is larger than the size of macroblocks of the AVC.

[0055]

In the following description, the "region" includes all of the various types of regions described above (for example, macroblocks, sub-macroblocks, LCUs, CUs, SCUs, PUs, TUs, and the like) (and may be any one of the regions). The "region" may naturally include units other than the above-described units, and units that are not usable depending on the content of description are appropriately excluded.

[0056]

Fig. 1 illustrates a configuration example of prediction units (PUs) which are the units of prediction process for CUs which are the units of encoding process.

As illustrated in Fig. 1, four types of PUs can be formed for one CU. The four large squares illustrated in Fig. 1 indicate CUs, and rectangles or squares inside the large square indicate PUs. Numbers indicate the index of each PU but do not indicate the content of an image.

[0057]

As illustrated in Fig. 1, in the example on the top-left corner, a CU includes one PU (2N × 2N). That is, in this case, the CU is equivalent to the PU. Moreover, in the example on the top-right corner, the CU is vertically split into two regions and includes two horizontally long PUs (2N × N). Further, in the example on the bottom-left corner, the CU is horizontally divided into two regions and includes two vertically long PUs (N × 2N). Moreover, in the example on the bottom-right corner, the CU is vertically and horizontally split into two regions (four regions in total) and includes four square PUs (N × N). A pattern that is to be used among these patterns is determined according to the content (the cost function value of prediction results) of an image.

[0058]

Non-Patent Document 3 proposes a method (merge mode) called Motion Partition Merging as one motion information encoding scheme. In this method, two flags, that is, MergeFlag and MergeLeftFlag, are transmitted as merge information which is the information on the merge mode. MergeFlag = 1 indicates that motion information of a current region (also referred to as a target region) X which is a processing target is the same as motion information of an upper neighboring region T adjacent to

the current region or a left neighboring region L adjacent to the current region. In this case, MergeLeftFlag is included in the merge information and is transmitted. MergeFlag = 0 indicates that the motion information of a current region X is different from the motion information of any one of the neighboring region T and the neighboring region L. In this case, the motion information of the current region X is transmitted.

[0059]

When the motion information of the current region X is the same as the motion information of the neighboring region L, MergeFlag = 1 and MergeLeftFlag = 1 are satisfied. When the motion information of the current region X is the same as the motion information of the neighboring region T, MergeFlag = 1 and MergeLeftFlag = 0 are satisfied.

[0060]

In such a merge mode, a temporally neighboring region (temporal correlation region) as well as the spatially neighboring regions L and T is considered to be used as a candidate region to be merged with the current region X.

[0061]

As illustrated in Fig. 2, the reference image indexes are determined as in the table illustrated in Fig. 3 based on the reference image indexes of a left neighboring region A, an upper neighboring region B, a top-right neighboring region C adjacent to a current region (current block) which is a target to be processed.

[0062]

In the table illustrated in Fig. 3, the second to

fourth columns from the left respectively indicate the states of the reference image indexes of the neighboring regions A to C. The first column from the left is the determined reference image index. "x," "y," and "z" indicate optional natural numbers and "-1" indicates that the neighboring region cannot be referred to.

[0063]

When there is only one region that can be referred to among the neighboring regions A to C, the reference image index of that block is used. Moreover, when there are two regions that can be referred to among the neighboring regions A to C, the smallest reference image index is used. Further, when all of the neighboring regions A to C cannot be referred to, the reference image index is set to 0.

[0064]

The temporal correlation region, located temporally around the current region which is a processing target, is determined as illustrated in Fig. 4. In Fig. 4, the left regions indicate partial regions of a current picture (CurrPic) (also referred to as a target picture) which is a target to be processed, and a top-left rectangle among the regions is a current region (CurrPU). Moreover, the right regions in Fig. 4 indicate partial regions of a temporal correlation picture (colPic) located temporally around the current picture. In this temporal correlation picture, a region including a pixel at the same position as the bottom-right pixel of the current region is a temporal correlation region (colPU). When this region is not referable, a region including a pixel at the same position as the central pixel of a

decoding region is set as the temporal correlation region (colPU).

[0065]

Moreover, the temporal correlation picture is determined as illustrated in Fig. 5. For example, when a current picture is a B-picture and collocated_from_l0_flag is "1," a picture indicated by a reference image index "0" of List L1 is used as a temporal correlation picture. Moreover, when a current picture is a P-picture or a B-picture and collocated_from_l0_flag is "0," a picture indicated by a reference image index "0" of List L0 is used as a temporal correlation picture.

[0066]

Further, a jump flag is set as illustrated in Fig. 6 according to a positional relationship between a current picture and a temporal correlation picture. For example, as illustrated on the upper side of Fig. 6, a temporal position of a temporal correlation picture in a reference image jumps over the current picture (the current picture is present between the temporal correlation picture and the reference image), the jump flag is set to "1."

[0067]

Moreover, a temporal position of a temporal correlation picture in a reference image does not jump over the current picture (the current picture is not present between the temporal correlation picture and the reference image), the jump flag is set to "0." When the jump flag is "1," since the current region becomes an interpolation between the temporal correlation region and

the reference image of the temporal correlation region, the reliability of the predictive vector is high.

[0068]

Moreover, although a motion vector mvCol of the temporal correlation region is used when generating a predictive vector pmv, in this case, the motion vector of the temporal correlation region is scaled similarly to the example illustrated in Fig. 7. That is, scaling is performed as in the following expressions (1) and (2) based on a temporal distance A between a current region and a reference image of the current region and a temporal distance B between a temporal correlation region and a reference image of the temporal correlation region.

[0069]

If A and B are identical, pmv = mvCol ··· (1)

If A and B are not identical, pmv = mvCol × (A/B) ··· (2)

[0070]

However, in the case of multi-view images, an image includes a plurality of views, and parallax prediction using correlation between views (in the parallax direction) is also performed. Fig. 8 illustrates an example of a reference relationship of a 3-view image.

[0071]

The 3-view image illustrated in Fig. 8 includes images of three views 0, 1, and 2. In Fig. 8, POC indicates the index of time. Moreover, PicNum indicates the index of decoding order.

[0072]

View 0 is called a base view and is encoded using temporal prediction that performs prediction using

temporal correlation. View 1 is called a non-base view and is encoded using temporal prediction and parallax prediction. In the parallax prediction, an encoded view 0 and a view 2 can be referred to. View 2 is called a non-base view and is encoded using temporal prediction and parallax prediction. In the parallax prediction, the encoded view 0 can be referred to.

[0073]

However, as described above, the conventional predictive vector relates to motion vectors only and encoding (prediction) of a parallax vector indicating a positional relationship between partial images that are the same or most similar between views, generated in parallax prediction across views has not been taken into consideration. The parallax vector is information that corresponds to a motion vector of temporal prediction and is used for temporal prediction that generates a predicted image of a current region using different images of different views of the same time. Thus, it may be not possible to predict parallax vectors appropriately and the encoding efficiency may decrease.

[0074]

Therefore, in the present technique, as described below, prediction of parallax vectors and motion vectors (motion parallax vectors) of a multi-view image is performed.

[0075]

For example, prediction of a parallax vector (also referred to as a current parallax vector) of a current region is performed similarly to the example illustrated in Fig. 9. In this example, a reference image of the

same view (view_id = 1) is allocated to a reference image index 0 of List 1, and a reference image of a different view (view_id = 2) is allocated to a reference image index 1.

[0076]

When the reference image index 0 (RefPicList[0]) of List 1 is used for a temporal correlation picture, the vector of a temporal correlation region (colPicB) included in the temporal correlation picture is employed as a predictive vector during a parallax vector that refers to different views of the same time.

[0077]

Moreover, when a reference image index 1 (RefPicList[1]) of List 1 is used for a view correlation picture, the vector of a view correlation region (colPicA) included in the view correlation picture is employed as a predictive vector during a parallax vector that refers to different view of the same time.

[0078]

That is, in the present technique, in order to predict a current parallax vector, all reference image indexes are used as candidates for a correlation picture. Moreover, it is determined whether the vector of each correlation region is a parallax vector similarly to the vector of the current region.

[0079]

Moreover, prediction of a motion vector (also referred to as a current motion vector) of a current region is performed similarly to the example illustrated in Fig. 10. In this example, a reference image of the same view (view_id = 1) is allocated to a reference image

index 0 of List 1, and a reference image of a different view (view_id = 2) is allocated to a reference image index 1.

[0080]

When the reference image index 0 (RefPicList[0]) of List 1 is used for a temporal correlation picture, the vector of a temporal correlation region (colPicB) included in the temporal correlation picture is employed as a predictive vector during a motion vector that refers to different times of the same view.

[0081]

Moreover, when a reference image index 1 (RefPicList[1]) of List 1 is used for a view correlation picture, the vector of a view correlation region (colPicA) included in the view correlation picture is employed as a predictive vector during a motion vector that refers to different times of the same view.

[0082]

That is, in the present technique, in order to predict a current motion vector, all reference image indexes are used as candidates for a correlation picture. Moreover, it is determined whether the vector of each correlation region is a motion vector similarly to the vector of the current region.

[0083]

Moreover, a scaling process when a parallax vector is used as a predictive vector is performed as follows. That is, the predictive vector is scaled based on an inter-view distance between the current region and the reference image thereof and an inter-view distance between the correlation region and the reference image

thereof.

[0084]

In the conventional technique, since only the motion vector is predicted, only a temporal distance is used. However, in the case of multi-view images, it is necessary to predict the parallax vector and thus the present technique also uses the inter-view distance. Accordingly, it is possible to improve the encoding efficiency.

[0085]

[Image encoding device]

Fig. 11 is a block diagram illustrating a main configuration example of an image encoding device which is an image processing device.

[0086]

An image encoding device 100 illustrated in Fig. 11 encodes image data using a prediction process similarly to the encoding scheme such as AVC or HEVC. However, the image encoding device 100 encodes a multi-view image including a plurality of view images. In the following description, a case where a 3-view image including three view images is processed as an example of a multi-view image will be described. However, actually, the image encoding device 100 can encode a multi-view image including an optional number of view points (views).

[0087]

As illustrated in Fig. 11, the image encoding device 100 includes an A/D converter 101, a screen rearrangement buffer 102, an arithmetic unit 103, an orthogonal transform unit 104, a quantization unit 105, a lossless encoding unit 106, and an accumulation buffer

107. Moreover, the image encoding device 100 includes an inverse quantization unit 108, an inverse orthogonal transform unit 109, an arithmetic unit 110, a loop filter 111, a decoded picture buffer 112, a selector 113, an intra-prediction unit 114, a motion parallax prediction/compensation unit 115, a predicted image selector 116, and a decoded multi-view picture buffer 121.

[0088]

The A/D converter 101 performs A/D conversion on input image data and supplies the converted image data (digital data) to the screen rearrangement buffer 102 which stores the image data. The screen rearrangement buffer 102 rearranges the frames of an image arranged in the stored order according to a GOP (Group Of Picture) so that the frames are rearranged in the order for encoding to obtain an image in which the frame order is rearranged and supplies the image to the arithmetic unit 103 together with the view ID and POC of the image.

[0089]

The screen rearrangement buffer 102 supplies the image in which the frame order is rearranged to the intra-prediction unit 114 and the motion parallax prediction/compensation unit 115 together with the view ID and POC of the image. The view ID is information for identifying a viewpoint and the POC is information for identifying the time.

[0090]

The arithmetic unit 103 subtracts a predicted image supplied from the intra-prediction unit 114 or the motion parallax prediction/compensation unit 115 via the predicted image selector 116 from the image read from the

screen rearrangement buffer 102 to obtain difference information thereof and outputs the difference information to the orthogonal transform unit 104.

[0091]

For example, in the case of an image which is subjected to intra-encoding, the arithmetic unit 103 subtracts the predicted image supplied from the intra-prediction unit 114 from the image read from the screen rearrangement buffer 102. Moreover, for example, in the case of an image which is subjected to inter-encoding, the arithmetic unit 103 subtracts the predicted image supplied from the motion parallax prediction/compensation unit 115 from the image read from the screen rearrangement buffer 102.

[0092]

The orthogonal transform unit 104 performs orthogonal transform such as a discrete cosine transform or a Karhunen-Loeve transform with respect to the difference information supplied from the arithmetic unit 103. The orthogonal transform method is optional. The orthogonal transform unit 104 supplies the transform coefficients to the quantization unit 105.

[0093]

The quantization unit 105 quantizes the transform coefficients supplied from the orthogonal transform unit 104. The quantization unit 105 sets quantization parameters based on the information on a target coding rate and performs quantization. The quantization method is optional. The quantization unit 105 supplies the quantized transform coefficients to the lossless encoding unit 106.

[0094]

The lossless encoding unit 106 encodes the transform coefficients quantized by the quantization unit 105 according to an optional encoding scheme. Moreover, the lossless encoding unit 106 acquires intra-prediction information including information or the like that indicates an intra-prediction mode from the intra-prediction unit 114 and acquires inter-prediction information including information that indicates an inter-prediction mode, motion parallax vector information, and the like from the motion parallax prediction/compensation unit 115. Further, the lossless encoding unit 106 acquires filter coefficients and the like used in the loop filter 111.

[0095]

The lossless encoding unit 106 encodes these various types of information according to an optional encoding scheme and incorporates (multiplexes) the information as part of the header information of the encoded data. The lossless encoding unit 106 supplies the encoded data obtained by encoding to the accumulation buffer 107 which accumulates the encoded data.

[0096]

Examples of the encoding scheme of the lossless encoding unit 106 include a variable-length encoding and an arithmetic encoding. An example of the variable-length encoding includes context-adaptive variable length coding (CAVLC) which is defined in the H.264/AVC scheme. An example of the arithmetic encoding includes context-adaptive binary arithmetic coding (CABAC).

[0097]

The accumulation buffer 107 temporarily stores the encoded data supplied from the lossless encoding unit 106. The accumulation buffer 107 outputs the encoded data stored therein to a recording device (recording medium) (not illustrated), a transmission line, and the like in the subsequent stage, for example, at a predetermined timing as a bit stream. That is, various items of encoded information are supplied to the decoding side.

[0098]

Moreover, the transform coefficients quantized in the quantization unit 105 are also supplied to the inverse quantization unit 108. The inverse quantization unit 108 performs inverse quantization on the quantized transform coefficients according to a method corresponding to the quantization of the quantization unit 105. The inverse quantization method is optional as long as the method corresponds to the quantization process of the quantization unit 105. The inverse quantization unit 108 supplies the obtained transform coefficients to the inverse orthogonal transform unit 109.

[0099]

The inverse orthogonal transform unit 109 performs inverse orthogonal transform on the transform coefficients supplied from the inverse quantization unit 108 according to a method corresponding to the orthogonal transform process of the orthogonal transform unit 104. The inverse orthogonal transform method is optional as long as the method corresponds to the orthogonal transform process of the orthogonal transform unit 104. The output (locally reconstructed difference information) obtained through the inverse orthogonal transform is

supplied to the arithmetic unit 110.

[0100]

The arithmetic unit 110 adds the predicted image supplied from the intra-prediction unit 114 or the motion parallax prediction/compensation unit 115 via the predicted image selector 116 to the inverse orthogonal transform result, that is, the locally reconstructed difference information, supplied from the inverse orthogonal transform unit 109 to obtain a locally reconstructed image (hereinafter referred to as a reconstructed image). The reconstructed image is supplied to the loop filter 111 or the decoded picture buffer 112.

[0101]

The loop filter 111 includes a deblocking filter, an adaptive loop filter, or the like and performs a filtering process appropriately with respect to the decoded image supplied from the arithmetic unit 110. For example, the loop filter 111 removes a block distortion of the decoded image by performing a deblocking filtering process on the decoded image. Moreover, for example, the loop filter 111 improves image quality by performing a loop filtering process using a Wiener filter on the deblocking filtering results (the decoded image in which the block distortion is removed).

[0102]

The loop filter 111 may perform an optional filtering process on the decoded image. Moreover, the loop filter 111 supplies information such as filter coefficients used for the filtering process to the lossless encoding unit 106 as necessary so that the

information is encoded.

[0103]

The loop filter 111 supplies the filtering result (hereinafter referred to as the decoded image) to the decoded picture buffer 112.

[0104]

The decoded picture buffer 112 stores the reconstructed image supplied from the arithmetic unit 110 and the decoded image supplied from the loop filter 111. Moreover, the decoded picture buffer 112 stores the view ID and POC of the image.

[0105]

The decoded picture buffer 112 supplies the reconstructed image (with the view ID and POC of the image) stored therein to the intra-prediction unit 114 via the selector 113 at a predetermined timing or based on a request from an external unit such as the intra-prediction unit 114. Moreover, the decoded picture buffer 112 supplies the decoded image (with the view ID and POC of the image) stored therein to the motion parallax prediction/compensation unit 115 via the selector 113 at a predetermined timing or based on a request from an external unit such as the motion parallax prediction/compensation unit 115.

[0106]

The selector 113 indicates a supply destination of the image output from the decoded picture buffer 112. For example, in the case of intra-prediction, the selector 113 reads the image (reconstructed image) that is not filtered from the decoded picture buffer 112 and supplies the image to the intra-prediction unit 114 as

neighboring pixels.

[0107]

Moreover, for example, in the case of inter-prediction, the selector 113 reads the filtered image (decoded image) from the decoded picture buffer 112 and supplies the image to the motion parallax prediction/compensation unit 115 as a reference image.

[0108]

Upon acquiring images (neighboring images) of neighboring regions located around a processing target region from the decoded picture buffer 112, the intra-prediction unit 114 performs intra-prediction (intra-field prediction) that generates a predicted image basically using a prediction unit (PU) as a processing unit, using the pixel values of the neighboring images. The intra-prediction unit 114 performs the intra-prediction in a plurality of modes (intra-prediction modes) prepared in advance.

[0109]

The intra-prediction unit 114 generates predicted images in all candidate intra-prediction modes, evaluates the cost function values of the respective predicted images using the input image supplied from the screen rearrangement buffer 102, and selects an optimal mode. When the optimal intra-prediction mode is selected, the intra-prediction unit 114 supplies the predicted image generated in the optimal mode to the predicted image selector 116.

[0110]

Moreover, the intra-prediction unit 114 supplies intra-prediction information including the information on

intra-prediction such as the optimal intra-prediction mode appropriately to the lossless encoding unit 106 which encodes the intra-prediction information.

[0111]

The motion parallax prediction/compensation unit 115 performs motion prediction and parallax prediction (inter-prediction) basically using PU as a processing unit, using the input image supplied from the screen rearrangement buffer 102 and the reference image supplied from the decoded picture buffer 112, performs a compensating process according to the detected motion parallax vector, and generates a predicted image (inter-prediction image information). The motion parallax prediction/compensation unit 115 performs such inter-prediction (inter-frame prediction) in a plurality of modes (inter-prediction modes) prepared in advance.

[0112]

The motion parallax prediction/compensation unit 115 generates predicted images in all candidate inter-prediction modes, evaluates cost function values of the respective predicted images, and selects an optimal mode. When an optimal inter-prediction mode is selected, the motion parallax prediction/compensation unit 115 supplies the predicted image generated in the optimal mode to the predicted image selector 116.

[0113]

Moreover, the motion parallax prediction/compensation unit 115 supplies inter-prediction information including the information on inter-prediction such as the optimal inter-prediction mode to the lossless encoding unit 106 which encodes the

inter-prediction information.

[0114]

The predicted image selector 116 selects a supplying source of the predicted image supplied to the arithmetic unit 103 and the arithmetic unit 110. For example, in the case of intra-encoding, the predicted image selector 116 selects the intra-prediction unit 114 as the supplying source of the predicted image and supplies the predicted image supplied from the intra-prediction unit 114 to the arithmetic unit 103 and the arithmetic unit 110. Moreover, for example, in the case of inter-encoding, the predicted image selector 116 selects the motion parallax prediction/compensation unit 115 as the supplying source of the predicted image and supplies the predicted image supplied from the motion parallax prediction/compensation unit 115 to the arithmetic unit 103 and the arithmetic unit 110.

[0115]

Although the decoded picture buffer 112 stores the image of a processing target view (with the view ID and POC of the image) only, the decoded multi-view picture buffer 121 stores the images of respective viewpoints (views) (with the view IDs and POCs of the images). That is, the decoded multi-view picture buffer 121 acquires the decoded image (with the view ID and POC of the image) supplied to the decoded picture buffer 112 and stores the decoded image (with the view ID and POC of the image) together with the decoded picture buffer 112.

[0116]

Although the decoded picture buffer 112 erases the decoded image when a processing target view changes, the

decoded multi-view picture buffer 121 stores the decoded image as it was. Moreover, the decoded multi-view picture buffer 121 supplies the stored decoded image (with the view ID and POC of the image) to the decoded picture buffer 112 as a "decoded image of a non-processing target view" according to a request of the decoded picture buffer 112 or the like. The decoded picture buffer 112 supplies the "decoded image of the non-processing target view (with the view ID and POC of the image)" read from the decoded multi-view picture buffer 121 to the motion parallax prediction/compensation unit 115 via the selector 113.

[0117]

[Motion parallax prediction/compensation unit]

Fig. 12 is a block diagram illustrating a main configuration example of the motion parallax prediction/compensation unit of Fig. 11.

[0118]

As illustrated in Fig. 12, the motion parallax prediction/compensation unit 115 includes a motion parallax vector search unit 131, a predicted image generating unit 132, an encoded information accumulation buffer 133, and a selector 134. Moreover, the motion parallax prediction/compensation unit 115 includes a spatial correlation predictive vector generating unit 135, a temporal parallax correlation predictive vector generating unit 136, a selector 137, an encoding cost calculating unit 138, and a mode determining unit 139.

[0119]

The motion parallax vector search unit 131 acquires a decoded image pixel value from the decoded picture

buffer 112 and acquires an original image pixel value from the screen rearrangement buffer 102. The motion parallax vector search unit 131 determines a reference image index of a current region which is a processing target using these values, performs motion search in the temporal direction and the parallax direction, and generates a current motion vector and a current parallax vector.

[0120]

In the following description, when it is not necessary to distinguish a motion vector indicating motion in the temporal direction (that is, between frames (pictures)) and a parallax vector indicating motion in the parallax direction (that is, between views) or both vectors are indicated, the vector(s) will be referred to as a motion parallax vector. A motion parallax vector of a current region is also referred to as a current motion parallax vector.

[0121]

The motion parallax vector search unit 131 supplies the reference image index and the motion parallax vector to the predicted image generating unit 132 and the encoding cost calculating unit 138.

[0122]

The predicted image generating unit 132 acquires the reference image index and the motion parallax vector from the motion parallax vector search unit 131 and acquires the decoded image pixel value from the decoded picture buffer 112. The predicted image generating unit 132 generates a predicted image of the current region using these values. The predicted image generating unit

132 supplies the predicted image pixel value to the encoding cost calculating unit 138.

[0123]

The encoded information accumulation buffer 133 stores mode information indicating the mode selected as the optimal mode in the mode determining unit 139 and the reference image index and the motion parallax vector of the mode. The encoded information accumulation buffer 133 supplies the stored information to the selector 134 at a predetermined timing or according to a request from an external unit.

[0124]

The selector 134 supplies the mode information, the reference image index, and the motion parallax vector supplied from the encoded information accumulation buffer 133 to the spatial correlation predictive vector generating unit 135 or the temporal parallax correlation predictive vector generating unit 136.

[0125]

The spatial correlation predictive vector generating unit 135 and the temporal parallax correlation predictive vector generating unit 136 generate a predictive value (predictive vector) of the motion vector (current motion vector) of the current region which is a processing target.

[0126]

The spatial correlation predictive vector generating unit 135 generates a predictive vector (spatial correlation predictive vector) using spatial correlation. More specifically, the spatial correlation predictive vector generating unit 135 acquires

information (the mode information, the reference image index, the motion parallax vector, and the like) on the motion information of a neighboring region (spatially neighboring region) located spatially around the current region, of the same frame (current frame (also referred to as a target frame)) as the current region from the encoded information accumulation buffer 133 via the selector 134.

[0127]

For example, the spatial correlation predictive vector generating unit 135 performs a median operation using the motion vectors (spatially neighboring motion vectors) of a plurality of spatially neighboring regions to generate a spatial correlation predictive vector. The spatial correlation predictive vector generating unit 135 supplies the generated spatial correlation predictive vector to the selector 137.

[0128]

The temporal parallax correlation predictive vector generating unit 136 generates a predictive vector (temporal parallax correlation predictive vector (temporal correlation predictive vector or parallax correlation predictive vector)) using temporal correlation or parallax correlation. More specifically, for example, the temporal parallax correlation predictive vector generating unit 136 acquires information on motion information of a neighboring region (temporally neighboring region) located temporally around the current region from the encoded information accumulation buffer 133 via the selector 134. The temporally neighboring region indicates a region (or the surrounding regions

thereof) located at the position corresponding to the current region of a frame (picture) different from the current frame, of the same view (current view (also referred to as a target view)) as the current region.

[0129]

Moreover, for example, the temporal parallax correlation predictive vector generating unit 136 acquires information on motion information of a neighboring region (parallactically neighboring region) located parallactically around the current region from the encoded information accumulation buffer 133 via the selector 134. The parallactically neighboring region indicates a region (or the surrounding regions thereof) located at the position corresponding to the current region of a frame (picture) of the same time as the current frame, of a view different from the view (current view) of the current region.

[0130]

For example, the temporal parallax correlation predictive vector generating unit 136 performs a median operation using the motion vectors (temporally neighboring motion vectors) of a plurality of temporally neighboring regions to generate a temporal correlation predictive vector. Moreover, for example, the temporal parallax correlation predictive vector generating unit 136 performs a median operation using the motion vectors (parallactically neighboring motion vectors) of a plurality of parallactically neighboring regions to generate a parallax correlation predictive vector.

[0131]

The temporal parallax correlation predictive vector

generating unit 136 supplies the temporal parallax correlation predictive vector, generated in this manner, to the selector 137.

[0132]

The spatial correlation predictive vector generating unit 135 and the temporal parallax correlation predictive vector generating unit 136 respectively generate the predictive vector in each inter-prediction mode.

[0133]

The selector 137 supplies the spatial correlation predictive vector supplied from the spatial correlation predictive vector generating unit 135 and the temporal parallax correlation predictive vector supplied from the temporal parallax correlation predictive vector generating unit 136 to the encoding cost calculating unit 138.

[0134]

The encoding cost calculating unit 138 calculates a difference value (difference image) between the predicted image and the original image in each inter-prediction mode using the predicted image pixel value supplied from the predicted image generating unit 132 and the original image pixel value supplied from the screen rearrangement buffer 102. Moreover, the encoding cost calculating unit 138 calculates a cost function value (also referred to as an encoding cost value) in each inter-prediction mode using the difference image pixel value.

[0135]

Further, the encoding cost calculating unit 138 selects a predictive vector that is closer to the motion

parallax vector of the current region supplied from the motion parallax vector search unit 131 among the spatial correlation predictive vector and the temporal parallax correlation predictive vector supplied from the selector 137 as the predictive vector of the current region. Moreover, the encoding cost calculating unit 138 generates a difference motion parallax vector which is a difference between the predictive vector and the motion parallax vector of the current region. The encoding cost calculating unit 138 generates the difference motion parallax vector in each inter-prediction mode.

[0136]

The encoding cost calculating unit 138 supplies the encoding cost value, the predicted image pixel value, and the difference motion parallax information including the difference motion parallax vector, of each inter-prediction mode, and prediction information including the predictive vector and the reference image index to the mode determining unit 139.

[0137]

The mode determining unit 139 selects an inter-prediction mode in which the encoding cost value is minimized as the optimal mode. The mode determining unit 139 supplies the predicted image pixel value of the inter-prediction mode, selected as the optimal mode, to the predicted image selector 116.

[0138]

When the inter-prediction is selected by the predicted image selector 116, the mode determining unit 139 supplies the mode information which is the information on the inter-prediction mode selected as the

optimal mode, the difference motion parallax information and the prediction information in the inter-prediction mode to the lossless encoding unit 106 which encodes the information. These items of information are encoded and transmitted to the decoding side.

[0139]

Moreover, the mode determining unit 139 supplies the mode information, the difference motion parallax information, and the prediction information in the inter-prediction mode selected as the optimal mode to the encoded information accumulation buffer 133 which stores the information. These items of information are used as information on the neighboring regions in a process for another region processed later than the current region.

[0140]

[Temporal parallax correlation predictive vector generating unit]

Fig. 13 is a block diagram illustrating a main configuration example of the temporal parallax correlation predictive vector generating unit 136.

[0141]

As illustrated in Fig. 13, the temporal parallax correlation predictive vector generating unit 136 includes a current region processor (target region processor) 151, a correlation region processor 152, an LI-prediction processor 153, an L0-prediction processor 154, a scheme-1 processor 155, a scheme-2 processor 156, a scheme-3 processor 157, a scheme-4 processor 158, and a predictive vector generating unit 159.

[0142]

The current region processor 151 performs a process

of acquiring the information on the current region. The current region processor 151 supplies the acquired information on the current region to the respective units ranging from the correlation region processor 152 to the L0-prediction processor 154. The correlation region processor 152 performs a process of acquiring the information on the correlation region.

[0143]

The correlation region is a region that is referred to in order to use correlation with the current region. For example, the temporal correlation region is a region that is referred to in order to use temporal correlation with the current region and is a temporally neighboring region having a motion vector used for generating the temporal correlation predictive vector. Moreover, the parallax correlation region is a region that is referred to in order to use parallax correlation with the current region and is a parallactically neighboring region having a motion vector used for generating the parallax correlation predictive vector. The correlation region includes these regions.

[0144]

The correlation region processor 152 supplies the information on the correlation region to the L1-prediction processor 153 and the L0-prediction processor 154.

[0145]

The L1-prediction processor 153 performs a prediction process in the L1 direction. The L1-prediction processor 153 acquires necessary information from the screen rearrangement buffer 102 and the decoded

picture buffer 112. Moreover, the L1-prediction processor 153 acquires the information supplied from the current region processor 151 and the correlation region processor 152. The L1-prediction processor 153 performs a prediction process in the L1 direction using these items of information.

[0146]

Four methods of schemes 1 to 4 are prepared as the prediction process. The L1-prediction processor selects any one of the methods and provides information to a processor corresponding to the method selected among the scheme-1 processor 155 to the scheme-4 processor 158.

[0147]

The L0-prediction processor 154 performs a prediction process in the L0 direction similarly to the L1-prediction processor 153.

[0148]

Scheme 1 is a scheme in which a frame (reference image) of the same time as the current frame, of a view different from the current view, allocated to the reference image index 1 of List 1 is used as a correlation image, and when the vector of the correlation region is a parallax vector, the parallax vector (reference parallax vector) is employed as a predictive vector. The scheme-1 processor 155 performs a process for generating the predictive vector according to such a scheme. The scheme-1 processor 155 supplies various parameters obtained through the process to the predictive vector generating unit 159.

[0149]

Scheme 2 is a scheme in which a frame (reference

image) of the time different from the current frame, of the same view as the current view, allocated to the reference image index 0 of List 1 is used as a correlation image, and when the vector of the correlation region is a parallax vector, the parallax vector (reference parallax vector) is employed as a predictive vector. The scheme-2 processor 156 performs a process for generating the predictive vector according to such a scheme. The scheme-2 processor 156 supplies various parameters obtained through the process to the predictive vector generating unit 159.

[0150]

Scheme 3 is a scheme in which a frame (reference image) of the time different from the current frame, of the same view as the current view, allocated to the reference image index 1 of List 1 is used as a correlation image, and when the vector of the correlation region is a motion vector, the motion vector (reference parallax vector) is employed as a predictive vector. The scheme-3 processor 157 performs a process for generating the predictive vector according to such a scheme. The scheme-3 processor 157 supplies various parameters obtained through the process to the predictive vector generating unit 159.

[0151]

Scheme 4 is a scheme in which a frame (reference image) of the time different from the current frame, of the same view as the current view, allocated to the reference image index 0 of List 1 is used as a correlation image, and when the vector of the correlation region is a motion vector, the motion vector (reference

parallax vector) is employed as a predictive vector. The scheme-4 processor 158 performs a process for generating the predictive vector according to such a scheme. The scheme-4 processor 158 supplies various parameters obtained through the process to the predictive vector generating unit 159.

[0152]

The predictive vector generating unit 159 generates a temporal parallax correlation predictive vector using the information supplied from the scheme-1 processor 155 to the scheme-4 processor 158 and the view information, the time information, and the like of the reference image acquired from the decoded picture buffer 112. In this case, the predictive vector generating unit 159 performs a scaling process using the information supplied from the scheme-1 processor 155 to the scheme-4 processor 158. In this case, the predictive vector generating unit 159 performs scaling in the temporal direction for the motion correlation predictive vector and performs scaling in the parallax direction for the parallax correlation predictive vector. The predictive vector generating unit 159 supplies the generated temporal parallax correlation predictive vector to the encoding cost calculating unit 138 via the selector 137.

[0153]

By doing so, the temporal parallax correlation predictive vector generating unit 136 can generate the parallax correlation predictive vector as well as the motion correlation predictive vector. Thus, the motion parallax prediction/compensation unit 115 can generate the predictive vector with high prediction accuracy even

when the vector of the current region is the parallax vector. Accordingly, the image encoding device 100 can suppress a decrease in the encoding efficiency.

[0154]

[Scheme selection example]

Fig. 14 illustrates examples in which each scheme is selected. As illustrated in Fig. 14, a region of which the positional relationship (whether a reference image is present in the temporal direction or in the parallax prediction) with the reference image is the same as the current region (target region) is selected as the correlation region. That is, the positional relationship between a current region and a reference image of the current region is identical to the positional relationship between the correlation region and a reference image of the correlation region. Moreover, the scheme is determined based on the positional relationship between the current region and the reference image of the current region and the positional relationship between the current region and the correlation region.

[0155]

In the table of Fig. 14, the first to fourth rows from the bottom indicate the example of the positional relationship of the respective images, and A to E indicate the example of the value of the view ID or the POC of each row. That is, what is important here is not how much the value is but whether the view ID or the POC of the image is identical to those of the other image.

[0156]

In the example of the fourth row from the bottom, both the current region and the correlation region have

different reference images and view IDs but have identical POCs. That is, the vectors of the current region and the correlation region are parallax vectors. Moreover, the current region and the correlation region have different view IDs and have identical POCs. That is, the correlation region is an image of a view of a different frame, of the same time as the current region. Thus, Scheme 1 is selected as illustrated in the table of Fig. 14. Scheme 1 is effective in a region in which a variation in the parallax between viewpoints is constant.

[0157]

In the example of the third row from the bottom, both the current region and the correlation region have different reference images and view IDs but have identical POCs. That is, the vectors of the current region and the correlation region are parallax vectors. Moreover, the current region and the correlation region have identical view IDs and have different POCs. That is, the correlation region is an image of a frame of a different time, of the same view as the current region. Thus, Scheme 2 is selected as illustrated in the table of Fig. 14. Scheme 2 is effective when a change in temporal motion is small.

[0158]

In the example of the second row from the bottom, both the current region and the correlation regions have identical reference images and view IDs but have different POCs. That is, the vectors of the current region and the correlation region are motion vectors. Moreover, the current region and the correlation region have different view IDs and have identical POCs. That is,

the correlation region is an image of a view of a different frame, of the same time as the current region. Scheme 3 is selected as illustrated in the table of Fig. 14. Scheme 3 is effective when a change in the parallax amount between viewpoints is small.

[0159]

In the example of the first row from the bottom, both the current region and the correlation region have identical reference images and view IDs but have different POCs. That is, the vectors of the current region and the correlation region are motion vectors. Moreover, the current region and the correlation region have identical view IDs and different POCs. That is, the correlation region is an image of a frame of a different time, of the same view as the current region. Thus, Scheme 4 is selected as illustrated in the table of Fig. 14.

[0160]

That is, for example, when any one (for example, the left-eye image) of the left and right images of a 3D image is a base view and the other (for example, the right-eye image) is a dependent view, and in the dependent view, the property of a vector (coding vector) of the current region is identical to the property of a vector (co-located vector) of a correlation region (co-located block) of a frame of a different time of the same view, the co-located block is set to be available. In other words, in the dependent view, when the properties of the coding vector and the co-located vector are not identical, the co-located block is set to be not available. Naturally, the same can be applied to the

base view.

[0161]

For example, the L1-prediction processor 153 and the L0-prediction processor 154 of Fig. 13 perform such setting.

[0162]

Whether or not the properties of the coding vector and the co-located vector are identical can be determined, for example, by comparing the POCs of the current region and the co-located block with the POCs of the respective reference images as illustrated in Fig. 15. For example, the L1-prediction processor 153 and the L0-prediction processor 154 of Fig. 13 perform such determination.

[0163]

For example, when the POC (CurrPOC) of the current region and the POC (CurrRefPOC) of the reference image of the current region are not identical and the POC (ColPOC) of the co-located block and the POC (ColRefPOC) of the reference image of the co-located block are not identical ((CurrPOC ! = CurrRefPOC)&&(ColPOC ! = ColRefPOC)), the L1-prediction processor 153 and the L0-prediction processor 154 determine that both the coding vector and the co-located block are motion vectors (A of Fig. 16).

[0164]

Moreover, for example, when the POC (CurrPOC) of the current region and the POC (CurrRefPOC) of the reference image of the current region are identical and the POC (ColPOC) of the co-located block and the POC (ColRefPOC) of the reference image of the co-located block are identical ((CurrPOC == CurrRefPOC)&&(ColPOC == ColRefPOC)), the L1-prediction processor 153 and the L0-

prediction processor 154 determine that both the coding vector and the co-located block are parallax vectors (inter-view vectors) (B of Fig. 16).

[0165]

The L1-prediction processor 153 and the L0-prediction processor 154 set the availability of the co-located vector as illustrated in the table on the lower side of Fig. 15 based on the determination results.

[0166]

For example, when both the coding vector and the co-located block are determined to be motion vectors or parallax vectors (inter-view vectors), the co-located vector is set to be available (A of Fig. 16 or B of Fig. 16).

[0167]

Conversely, when one of the coding vector and the co-located block is determined to be the motion vector and the other is determined to be the parallax vector (inter-view vector), the co-located vector is set to be not available.

[0168]

[Syntax]

Fig. 17 illustrates an example of the syntax of the sequence parameter set of this case. As illustrated on the tenth row to the third row from the bottom of Fig. 17, information such as a total number of views, an ID for identifying views, the number of parallax predictions in List L0, an ID of a view referred to in the parallax prediction in List L0, the number of parallax predictions in List L1, and an ID of a view referred to in the parallax prediction in List L1 is included in the

sequence parameter set. These items of information are information necessary for multi-view images. In other words, the present technique can be applied without adding new syntax to the sequence parameter set.

[0169]

Fig. 18 illustrates an example of the syntax of a slice header of this case. As illustrated on the eighth row from the bottom of Fig. 18, an ID for identifying views is included in the slice header. This information is information necessary for multi-view images. In other words, the present technique can be applied without adding new syntax to the slice header.

[0170]

Fig. 19 illustrates an example of the syntax of a prediction unit of this case. As illustrated in Fig. 19, the present technique can be applied without adding new syntax to the prediction unit. However, since the application of the present technique increases the number of candidate correlation regions as compared to the conventional technique, it is necessary to expand the type of the syntax or change the content of the process for the merge mode ID and predictive vector ID.

[0171]

[Flow of encoding process]

Next, the flow of the respective processes executed by the image encoding device 100 having such a configuration will be described. First, an example of the flow of the encoding process will be described with reference to the flowchart of Fig. 20.

[0172]

In step S101, the A/D converter 101 performs A/D

conversion on an input image. In step S102, the screen rearrangement buffer 102 stores the A/D-converted image and rearranges the respective pictures so that the pictures arranged in the display order is rearranged in the encoding order.

[0173]

In step S103, the intra-prediction unit 114 performs an intra-prediction process. In step S104, the motion parallax prediction/compensation unit 115 performs an inter-motion prediction process. In step S105, the predicted image selector 116 selects any one of the predicted image generated by the intra-prediction and the predicted image generated by the inter-prediction.

[0174]

In step S106, the arithmetic unit 103 calculates (generates a difference image) a difference between the image rearranged by the process of step S102 and the predicted image selected by the process of step S105. The generated difference image has a smaller data amount than the original image. Thus, it is possible to compress the data amount as compared to when the image is encoded as it is.

[0175]

In step S107, the orthogonal transform unit 104 performs orthogonal transform on the difference image generated by the process of step S106. Specifically, orthogonal transform such as a discrete cosine transform or a Karhunen-Loeve transform is performed, and orthogonal transform coefficients are output. In step S108, the quantization unit 105 quantizes the orthogonal transform coefficients obtained by the process of step

S107.

[0176]

The difference image quantized by the process of step S108 is locally decoded in the following manner. That is, in step S109, the inverse quantization unit 108 performs inverse quantization on the quantized orthogonal transform coefficients (also referred to as quantization coefficients) generated by the process of step S108 according to a property corresponding to the property of the quantization unit 105. In step S110, the inverse orthogonal transform unit 109 performs inverse orthogonal transform on the orthogonal transform coefficient obtained by the process of step S109 according to a property corresponding to the property of the orthogonal transform unit 104. In this manner, the difference image is reconstructed.

[0177]

In step S111, the arithmetic unit 110 adds the predicted image selected in step S105 to the difference image generated in step S110 to generate a locally decoded image (reconstructed image). In step S112, the loop filter 111 performs a loop filtering process including a deblocking filtering process or an adaptive loop filtering process appropriately with respect to the reconstructed image obtained by the process of step S111 to generate a decoded image.

[0178]

In step S113, the decoded picture buffer 112 and the decoded multi-view picture buffer 121 store the decoded image generated by the process of step S112 and the reconstructed image generated by the process of step

S111.

[0179]

In step S114, the lossless encoding unit 106 encodes the orthogonal transform coefficients quantized by the process of step S108. That is, lossless encoding such as variable-length encoding or arithmetic encoding is performed with respect to the difference image. The lossless encoding unit 106 encodes information on prediction, information on quantization, information on the filtering process, and the like and adds the encoded information to a bit stream.

[0180]

In step S115, the accumulation buffer 107 accumulates the bit stream obtained by the process of step S114. The encoded data accumulated in the accumulation buffer 107 is appropriately read and is transmitted to the decoding side via a transmission line or a recording medium.

[0181]

In step S116, the quantization unit 105 controls the rate of the quantization operation based on the coding rate (occurrence coding rate) of the encoded data accumulated in the accumulation buffer 107 by the process of step S115 so that an overflow or an underflow does not occur.

[0182]

When the process of step S116 ends, the encoding process ends.

[0183]

[Flow of inter-motion prediction process]

Next, an example of the flow of an inter-motion

prediction process executed in step S104 of Fig. 20 will be described with reference to the flowchart of Fig. 21.

[0184]

In step S131, the motion parallax vector search unit 131 performs motion search with respect to the inter-prediction mode of a processing target to generate a motion parallax vector (motion vector or parallax vector) of the current region which is the processing target. In step S132, the predicted image generating unit 132 performs a compensation process using the motion parallax vector generated in step S131 to generate a predicted image. In step S133, the encoding cost calculating unit 138 generates a difference image between the predicted image generated in step S132 and the original image (input image).

[0185]

In step S134, the encoding cost calculating unit 138 performs a merge mode process using the spatial correlation predictive vector generating unit 135, the temporal parallax correlation predictive vector generating unit 136, and the like.

[0186]

In step S135, the encoding cost calculating unit 138 compares the motion parallax vector of the current region generated in step S131 and the predictive vector of the current region generated by the process of step S134 to determine whether a merge mode is to be applied to the current region.

[0187]

When it is determined that both are not identical and the merge mode is not to be applied, the encoding

cost calculating unit 138 proceeds to the process of step S136 and performs the parallax motion vector prediction process using the spatial correlation predictive vector generating unit 135, the temporal parallax correlation predictive vector generating unit 136, and the like. When the process of step S136 ends, the encoding cost calculating unit 138 proceeds to the process of step S137.

[0188]

Moreover, in step S135, when it is determined that the motion parallax vector and the predictive vector of the current region are identical and the merge mode is to be applied to the current region, the encoding cost calculating unit 138 skips the process of step S136 and proceeds to step S137.

[0189]

In step S137, the encoding cost calculating unit 138 determines whether the above process has been performed in all inter-prediction modes. When it is determined that a non-processed inter-prediction mode is present, the flow returns to step S131, and control is performed so that the subsequent processes are repeatedly performed with respect to the non-processed inter-prediction mode. That is, the processes of steps S131 to S137 are executed in the respective inter-prediction modes.

[0190]

When it is determined that the process has been performed in all inter-prediction modes in step S137, the encoding cost calculating unit 138 proceeds to the process of step S138. In step S138, the encoding cost calculating unit 138 calculates the cost function value

of each inter-prediction mode.

[0191]

In step S139, the mode determining unit 139 determines an inter-prediction mode in which the cost function value (encoding cost value) calculated in step S138 is the smallest as an optimal mode (optimal inter-prediction mode).

[0192]

In step S140, the predicted image generating unit 132 generates the predicted image in the optimal inter-prediction mode. The predicted image is supplied to the predicted image selector 116.

[0193]

In step S141, the encoded information accumulation buffer 133 stores the mode information and the motion information (the motion parallax vector, the reference image index, and the like) of the optimal inter-prediction mode when the inter-prediction is selected in step S105 of Fig. 20. When the intra-prediction mode is selected, a zero vector is stored as the motion parallax vector. When the inter-prediction is selected in step S105 of Fig. 20, these items of information are supplied to and encoded by the lossless encoding unit 106 and the encoded information is transmitted to the decoding side.

[0194]

When the process of step S141 ends, the encoded information accumulation buffer 133 ends the inter-motion prediction process and the flow proceeds to the flowchart of Fig. 20.

[0195]

[Flow of merge mode process]

Next, an example of the flow of the merge mode process executed in step S134 of Fig. 21 will be described with reference to the flowchart of Fig. 22.

[0196]

When the merge mode process starts, in step S161, the spatial correlation predictive vector generating unit 135 performs a spatial correlation prediction process of generating a spatial correlation predictive vector using the correlation with a spatially neighboring region. In step S162, the temporal parallax correlation predictive vector generating unit 136 performs a temporal correlation prediction process of generating a temporal parallax correlation predictive vector using the correlation with a temporally neighboring region or a parallactically neighboring region.

[0197]

In step S163, the encoding cost calculating unit 138 removes an overlapping vector from the spatial correlation predictive vector generated in step S161 and the temporal parallax predictive vector generated in step S162.

[0198]

In step S164, the encoding cost calculating unit 138 determines whether a vector is present. When it is determined that there is at least one spatial correlation predictive vector or temporal parallax correlation predictive vector, the encoding cost calculating unit 138 proceeds to the process of step S165.

[0199]

In step S165, the encoding cost calculating unit 138 determines whether a plurality of vectors is present.

When it is determined that a plurality of vectors is present, the encoding cost calculating unit 138 proceeds to the process of step S166 to acquire a merge index. When it is determined that a plurality of vectors is not present, the encoding cost calculating unit 138 skips the process of step S166.

[0200]

When a spatial correlation predictive vector or a temporal parallax correlation predictive vector identical to the motion vector of the current region is present, the encoding cost calculating unit 138 acquires the identical vector as a predictive vector in step S167 and acquires the reference image index in step S168.

[0201]

When the process of step S168 ends, the encoding cost calculating unit 138 ends the merge mode process and the flow returns to the flowchart of Fig. 21.

[0202]

Moreover, when it is determined that neither the spatial correlation predictive vector nor the temporal parallax correlation predictive vector is present in step S164, the encoding cost calculating unit 138 proceeds to the process of step S169.

[0203]

In step S169, the encoding cost calculating unit 138 assigns an initial value (for example, a zero vector) to the predictive vector. Moreover, in step S170, the encoding cost calculating unit 138 assigns an initial value (for example, 0) to the reference image index.

[0204]

When the process of step S170 ends, the encoding

cost calculating unit 138 ends the merge mode process and the flow returns to the flowchart of Fig. 21.

[0205]

[Flow of parallax motion vector prediction process]

Next, an example of the flow of the parallax motion vector prediction process executed in step S136 of Fig. 21 will be described with reference to the flowchart of Fig. 23.

[0206]

When the parallax motion vector prediction process starts, in step S191, the spatial correlation predictive vector generating unit 135 performs a spatial correlation prediction process to generate a spatial correlation predictive vector. In step S192, the temporal parallax correlation predictive vector generating unit 136 performs a temporal parallax correlation prediction process to generate a temporal parallax correlation predictive vector.

[0207]

In step S193, the encoding cost calculating unit 138 removes an overlapping vector from the spatial correlation predictive vector generated in step S191 and the temporal parallax predictive vector generated in step S192.

[0208]

The encoding cost calculating unit 138 selects a vector closest to the motion vector of the current region among the remaining vectors, creates a predictive vector index indicating the vector in step S194, and uses the vector as a predictive vector in step S195. When the process of step S195 ends, the encoding cost calculating

unit 138 ends the parallax motion vector prediction process and the flow returns to the flowchart of Fig. 21.

[0209]

[Flow of temporal parallax correlation prediction process]

Next, an example of the flow of the temporal parallax correlation prediction process executed in step S162 of Fig. 22 or step S192 of Fig. 23 will be described with reference to the flowchart of Figs. 24 to 27.

[0210]

When the temporal parallax correlation prediction process starts, the current region processor 151 acquires the view ID and POC of the current region in step S211 of Fig. 24. In step S212, the current region processor 151 determines a reference index of the current region. In step S213, the current region processor 151 acquires the view ID and POC of the reference image.

[0211]

In step S214, the correlation region processor 152 selects the correlation image in ascending order of the reference image indexes of List L1. In step S215, the correlation region processor 152 determines whether all reference image indexes have been searched. When it is determined that all reference image indexes have been searched, the temporal parallax correlation prediction process ends, and the flow returns to the flowchart of Fig. 22 or Fig. 23.

[0212]

Moreover, when it is determined in step S215 that a non-processed reference image index is present, the correlation region processor 152 proceeds to the process

of step S216. In step S216, the correlation region processor 152 determines a correlation region and the flow proceeds to the flowchart of Fig. 25.

[0213]

In step S221 of Fig. 25, the correlation region processor 152 determines whether the correlation region is an intra-predicted region or a region that does not refer to another region. When it is determined that the correlation region is the intra-predicted region or the region that does not refer to another region, the correlation region processor 152 proceeds to the flowchart of Fig. 26.

[0214]

Moreover, when it is determined in step S221 of Fig. 25 that the correlation region is the inter-predicted region and the region that does not refer to another region, the correlation region processor 152 proceeds to the process of step S222.

[0215]

In step S222, the L1-prediction processor 153 acquires the view ID and POC of the correlation region. In step S223, the L1-prediction processor 153 acquires the view ID and POC of the L1-prediction reference image of the correlation region.

[0216]

In step S224, the L1-prediction processor 153 determines whether the current region and the correlation region have identical view IDs. When both are identical, the correlation region is a co-located block. When it is determined that the current region and the correlation region have identical view IDs, the L1-prediction

processor 153 proceeds to the process of step S225 to perform a scheme-4-2 process so that the predictive vector is generated according to Scheme 4 or 2. When the process of step S225 ends, the flow proceeds to the flowchart of Fig. 26.

[0217]

Moreover, when it is determined in step S224 of Fig. 25 that the current region and the correlation region do not have identical view IDs, the L1-prediction processor 153 proceeds to the process of step S226.

[0218]

In step S226, the L1-prediction processor 153 determines whether the current region and the correlation region have identical POCs. When it is determined that the current region and the correlation region do not have identical POCs, the L1-prediction processor 153 proceeds to the flowchart of Fig. 26.

[0219]

Moreover, when it is determined in step S226 of Fig. 25 that the current region and the correlation region have identical POCs, the L1-prediction processor 153 proceeds to the process of step S227.

[0220]

In step S227, the L1-prediction processor 153 determines whether the reference image of the current region and the reference image of the correlation region have identical POCs. When it is determined that the reference image of the current region and the reference image of the correlation region have identical POCs, the L1-prediction processor 153 proceeds to the process of step S228 to perform a scheme-1 process so that the

predictive vector is generated according to Scheme 1. When the process of step S228 ends, the L1-prediction processor 153 proceeds to the flowchart of Fig. 26.

[0221]

Moreover, when it is determined in step S227 of Fig. 25 that the reference image of the current region and the reference image of the correlation region do not have identical POCs, the L1-prediction processor 153 proceeds to the process of step S229 to perform a scheme-3 process so that the predictive vector is generated according to Scheme 3. When the process of step S229 ends, the L1-prediction processor 153 proceeds to the flowchart of Fig. 26.

[0222]

In Fig. 26, the L0-prediction processor 154 performs the same process as the L1 prediction of Fig. 25 with respect to the L0 prediction of the correlation region. That is, the L0-prediction processor 154 performs the same processes of steps S231 to S238 as the processes of steps S222 to S229 of Fig. 25.

[0223]

However, in the case of Fig. 26, the reference image of the correlation region is a L0-prediction reference image. Moreover, when the process of step S234, S237, or S238 ends, or when it is determined in step S235 that the current region and the correlation region do not have identical POCs, the L0-prediction processor 154 proceeds to the flowchart of Fig. 27.

[0224]

In step S241 of Fig. 27, the predictive vector generating unit 159 determines whether at least one

predictive vector candidate is present. When it is determined that no predictive vector candidate is present, the predictive vector generating unit 159 returns to the process of step S214 of Fig. 24.

[0225]

Moreover, when it is determined in step S241 of Fig. 27 that at least one predictive vector candidate is present, the predictive vector generating unit 159 proceeds to the process of step S242. In step S242, the predictive vector generating unit 159 determines whether the jump flag of L1 only is 1 or whether the jump flags of L0 or L1 are identical, and L1 is referred to.

[0226]

When it is determined that the jump flag of L1 only is 1 or that the jump flags of L0 and L1 are identical, and L1 is referred to, the predictive vector generating unit 159 proceeds to the process of step S243.

[0227]

In step S243, the predictive vector generating unit 159 acquires an L1-reference vector. In step S244, the predictive vector generating unit 159 acquires a reference image index of L1 reference and the flow proceeds to step S247.

[0228]

Moreover, when it is determined in step S242 that the jump flag of L1 is not 1 or the jump flag of L0 is 0, and that the jump flags of L0 and L1 are not identical or L1 is not referred to, the predictive vector generating unit 159 proceeds to the process of step S245.

[0229]

In step S245, the predictive vector generating unit

159 acquires an L0-reference vector. In step S246, the predictive vector generating unit 159 acquires a reference image index of L0 reference and the flow proceeds to step S247.

[0230]

In step S247, the predictive vector generating unit 159 uses the L1-reference vector or L0-reference vector acquired as the predictive vector. In this case, the predictive vector generating unit 159 scales the vector using the distance to the reference image in the current region or the correlation region and uses the scaling result as the predictive vector.

[0231]

When the process of step S247 ends, the predictive vector generating unit 159 returns to the flowchart of Fig. 24 to end the temporal parallax correlation prediction process, and the flow returns to the flowchart of Fig. 22 or Fig. 23.

[0232]

[Flow of scheme-1 process]

Next, an example of the flow of the scheme-1 process executed in Fig. 25 or Fig. 26 will be described with reference to the flowchart of Fig. 28.

[0233]

When the scheme-1 process starts, in step S261, the scheme-1 processor 155 determines whether the POCs of the current region and the reference image of the current region are identical. When it is determined that the POCs of the current region and the reference image of the current region are not identical, the scheme-1 processor 155 ends the scheme-1 process and the flow returns to the

flowchart of Fig. 25 or Fig. 26.

[0234]

Moreover, when it is determined in step S261 that the POCs of the current region and the reference image of the current region are identical, the scheme-1 processor 155 proceeds to the process of step S262.

[0235]

In step S262, the scheme-1 processor 155 determines whether the POCs of the correlation region and the reference image of the correlation region are identical. When it is determined that the POCs of the correlation region and the reference image of the correlation region are not identical, the scheme-1 processor 155 ends the scheme-1 process and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0236]

Moreover, when it is determined in step S262 that the POCs of the correlation region and the reference image of the correlation region are identical, the scheme-1 processor 155 proceeds to the process of step S263.

[0237]

In step S263, the scheme-1 processor 155 calculates an inter-view distance 1 between the current region and the reference image of the current region. In step S264, the scheme-1 processor 155 calculates an inter-view distance 2 between the correlation region and the reference image of the correlation region. In step S265, the scheme-1 processor 155 determines the jump flag of the reference image, and the flow returns to the flowchart of Fig. 25 or Fig. 26. These parameters (the

inter-view distance 1, the inter-view distance 2, and the jump flag) are used in the process of Fig. 27.

[0238]

[Flow of scheme-3 process]

Next, an example of the flow of the scheme-3 process executed in Fig. 25 or Fig. 26 will be described with reference to the flowchart of Fig. 29.

[0239]

When the scheme-3 process starts, in step S271, the scheme-3 processor 157 determines whether the view IDs of the current region and the reference image of the current region are identical. When the view IDs of the current region and the reference image of the current region are not identical, the scheme-3 processor 157 ends the scheme-3 process, and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0240]

Moreover, when it is determined in step S271 that the view IDs of the current region and the reference image of the current region are identical, the scheme-3 processor 157 proceeds to the process of step S272.

[0241]

In step S272, the scheme-3 processor 157 determines whether the view IDs of the correlation region and the reference image of the correlation region are identical. When it is determined that the view IDs of the correlation region and the reference image of the correlation region are not identical, the scheme-3 processor 157 ends the scheme-3 process, and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0242]

Moreover, when it is determined in step S272 that the view IDs of the correlation region and the reference image of the correlation region are identical, the scheme-3 processor 157 proceeds to the process of step S273.

[0243]

In step S273, the scheme-3 processor 157 calculates the inter-view distance 1 between the current region and the reference image of the current region. In step S274, the scheme-3 processor 157 calculates the inter-view distance 2 between the correlation region and the reference image of the correlation region. In step S275, the scheme-3 processor 157 determines the jump flag of the reference image and the flow returns to the flowchart of Fig. 25 or Fig. 26. These parameters (the inter-view distance 1, the inter-view distance 2, and the jump flag) are used in the process of Fig. 27.

[0244]

[Flow of scheme-4-2 process]

Next, an example of the flow of the scheme-4-2 process executed in Fig. 25 or Fig. 26 will be described with reference to the flowchart of Fig. 30.

[0245]

When the scheme-4-2 process starts, in step S281, the L1-prediction processor 153 or the L0-prediction processor 154 determines whether the view IDs of the current region and the reference image of the current region are identical. When both are identical, a coding vector is a motion vector. When it is determined that the view IDs of the current region and the reference image of the current region are identical, the flow

proceeds to step S282.

[0246]

In step S282, the L1-prediction processor 153 or the L0-prediction processor 154 determines whether the view IDs of the correlation region (co-located block) and the reference image of the correlation region are identical. When both are not identical, a co-located vector is a parallax vector. When it is determined that the view IDs of the correlation region and the reference image of the correlation region are not identical, the L1-prediction processor 153 or the L0-prediction processor 154 ends the scheme-4-2 process and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0247]

That is, in this case, since the coding vector is the motion vector and the co-located vector is the parallax vector, the property of the coding vector is not identical to the property of the co-located vector. Thus, the co-located block is set to be not available, and the scheme-4-2 process ends.

[0248]

Moreover, when it is determined in step S282 that the view IDs of the correlation region and the reference image of the correlation region are identical, the L1-prediction processor 153 or the L0-prediction processor 154 proceeds to the process of step S283. When both are identical, the co-located vector is the motion vector. That is, in this case, both the encoding vector and the co-located vector are the motion vector, and the properties of both vectors are identical. Thus, in this case, the co-located vector is set to be available.

[0249]

In step S283, the scheme-4 processor 158 calculates the inter-view distance 1 between the current region and the reference image of the current region. In step S284, the scheme-4 processor 158 calculates the inter-view distance 2 between the correlation region and the reference image of the correlation region. In step S285, the scheme-4 processor 158 determines the jump flag of the reference image, and the flow returns to the flowchart of Fig. 25 or Fig. 26. These parameters (the inter-view distance 1, the inter-view distance 2, and the jump flag) are used in the process of Fig. 27.

[0250]

Moreover, when it is determined in step S281 that the view IDs of the current region and the reference image of the current region are not identical, the L1-prediction processor 153 or the L0-prediction processor 154 proceeds to the process of step S286.

[0251]

In step S286, the prediction processor 153 or the L0-prediction processor 154 determines whether the POCs of the current region and the reference image of the current region are identical. When it is determined that the POCs of the current region and the reference image of the current region are not identical, the prediction processor 153 or the L0-prediction processor 154 ends the scheme-4-2 process, and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0252]

Moreover, when it is determined in step S286 that the POCs of the current region and the reference image of

the current region are identical, the prediction processor 153 or the L0-prediction processor 154 proceeds to the process of step S287. When both are identical, a coding vector is a parallax vector.

[0253]

In step S287, the prediction processor 153 or the L0-prediction processor 154 determines whether the POCs of the correlation region and the reference image of the correlation region are identical. When both are not identical, a co-located vector is a motion vector. When it is determined that the POCs of the correlation region and the reference image of the correlation region are not identical, the prediction processor 153 or the L0-prediction processor 154 ends the scheme-4-2 process, and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0254]

That is, in this case, since the coding vector is the parallax vector and the co-located vector is the motion vector, the property of the coding vector is not identical to the property of the co-located vector. Thus, the co-located block is set to be not available, and the scheme-4-2 process ends.

[0255]

Moreover, when it is determined in step S287 that the POCs of the correlation region and the reference image of the correlation region are identical, the L1-prediction processor 153 or the L0-prediction processor 154 returns to the process of step S283. When both are identical, the co-located vector is the parallax vector. That is, in this case, both the coding vector and the co-located vector are parallax vectors, and the properties

of both vectors are identical. Thus, in this case, the co-located vector is set to be available.

[0256]

In this case, the scheme-2 processor 156 performs the processes of steps S283 to S285 similarly to the scheme-4 processor 158. When the process of step S285 ends, the scheme-2 processor 156 ends the schemme-4-2 process, and the flow returns to the flowchart of Fig. 25 or Fig. 26.

[0257]

By performing the respective processes in this manner, the temporal parallax correlation predictive vector generating unit 136 can generate the parallax correlation predictive vector as well as the motion correlation predictive vector. Thus, the motion parallax prediction/compensation unit 115 can generate the predictive vector with high prediction accuracy even when the vector of the current region is the parallax vector. Due to this, the image encoding device 100 can suppress a decrease in the encoding efficiency.

[0258]

[2. Second embodiment]

[Image decoding device]

Fig. 31 is a block diagram illustrating a main configuration example of an image decoding device which is an image processing device. An image decoding device 300 illustrated in Fig. 31 is a device that corresponds to the image encoding device 100 of Fig. 11. That is, the image decoding device 300 decodes the encoded data (bit stream), which is generated through encoding a multi-view image by the image encoding device 100,

according to a decoding method corresponding to the encoding method of the image encoding device 100 to obtain a decoded multi-view image.

[0259]

As illustrated in Fig. 31, the image decoding device 300 includes an accumulation buffer 301, a lossless decoding unit 302, an inverse quantization unit 303, an inverse orthogonal transform unit 304, an arithmetic unit 305, a loop filter 306, a screen rearrangement buffer 307, and a D/A converter 308. Moreover, the image decoding device 300 includes a decoded picture buffer 309, a selector 310, an intra-prediction unit 311, a motion parallax compensation unit 312, and a selector 313.

[0260]

Further, the image decoding device 300 includes a decoded multi-view picture buffer 321.

[0261]

The accumulation buffer 301 accumulates the encoded data transmitted thereto and supplies the encoded data to the lossless decoding unit 302 at a predetermined timing. The lossless decoding unit 302 decodes the information encoded by the lossless encoding unit 106 of Fig. 11, which is supplied from the accumulation buffer 301, according to a scheme corresponding to the encoding scheme of the lossless encoding unit 106. The lossless decoding unit 302 supplies quantized coefficient data of the difference image obtained by decoding to the inverse quantization unit 303.

[0262]

Moreover, the lossless decoding unit 302 refers to

the information on the optimal prediction mode obtained by decoding the encoded data to determine whether the intra-prediction mode or the inter-prediction mode is selected as the optimal prediction mode. The lossless decoding unit 302 supplies the information on the optimal prediction mode to the intra-prediction unit 311 or the motion parallax compensation unit 312 based on the determination result. That is, for example, when the image encoding device 100 selected the intra-prediction mode as the optimal prediction mode, the intra-prediction information which is the information on the optimal prediction mode or the like is supplied to the intra-prediction unit 311. Moreover, for example, when the image encoding device 100 selected the inter-prediction mode as the optimal prediction mode, the inter-prediction information which is the information on the optimal prediction mode or the like is supplied to the motion parallax compensation unit 312.

[0263]

The inverse quantization unit 303 performs inverse quantization on the quantized coefficient data, which is obtained through decoding by the lossless decoding unit 302, according to a scheme corresponding to the quantization scheme of the quantization unit 105 of Fig. 11 and supplies the obtained coefficient data to the inverse orthogonal transform unit 304. The inverse orthogonal transform unit 304 performs inverse orthogonal transform on the coefficient data supplied from the inverse quantization unit 303 according to a scheme corresponding to the orthogonal transform scheme of the orthogonal transform unit 104 of Fig. 11. The inverse

orthogonal transform unit 304 obtains the difference image corresponding to the difference image before being subjected to the orthogonal transform in the image encoding device 100 by the inverse orthogonal transform process.

[0264]

The difference image obtained through inverse orthogonal transform is supplied to the arithmetic unit 305. Moreover, the predicted image from the intra-prediction unit 311 or the motion parallax compensation unit 312 is supplied to the arithmetic unit 305 via the selector 313.

[0265]

The arithmetic unit 305 adds the difference image and the predicted image to obtain a reconstructed image corresponding to the image before subtraction of the predicted image by the arithmetic unit 103 of the image encoding device 100. The arithmetic unit 305 supplies the reconstructed image to the loop filter 306.

[0266]

The loop filter 306 performs a loop filtering process including a deblocking filtering process, an adaptive loop pressure filter, and the like with respect to the supplied reconstructed image to generate a decoded image. For example, the loop filter 306 removes a block distortion by performing a deblocking filtering process on the reconstructed image. Moreover, for example, the loop filter 306 improves image quality by performing a loop filtering process using a Wiener filter with respect to the deblocking filtering process results (the reconstructed image in which the block distortion is

removed).

[0267]

The type of the filtering process performed by the loop filter 306 is optional, and a filtering process other than the above-described processes may be performed. Moreover, the loop filter 306 may perform the filtering process using the filter coefficients supplied from the image encoding device 100 of Fig. 11.

[0268]

The loop filter 306 supplies the decoded image which is the filtering process result to the screen rearrangement buffer 307 and the decoded picture buffer 309. The filtering process of the loop filter 306 may be omitted. That is, the output of the arithmetic unit 305 may be stored in the decoded picture buffer 309 without being subjected to the filtering process. For example, the intra-prediction unit 311 uses the pixel values of pixels included in this image as the pixel values of the neighboring pixels.

[0269]

The screen rearrangement buffer 307 rearranges the supplied decoded image. That is, the frame order rearranged for the encoding order by the screen rearrangement buffer 102 of Fig. 11 is rearranged in the original display order. The D/A converter 308 performs D/A conversion on the decoded image supplied from the screen rearrangement buffer 307, outputs the converted image to a display (not illustrated), and displays the image.

[0270]

The decoded picture buffer 309 stores the supplied

reconstructed image (and the view ID and POC of the image) and the decoded image (and the view ID and POC of the image). Moreover, the decoded picture buffer 309 supplies the stored reconstructed image (and the view ID and POC of the image) or decoded image (and the view ID and POC of the image) to the intra-prediction unit 311 and the motion parallax compensation unit 312 via the selector 310 at a predetermined timing or based on a request of an external unit such as the intra-prediction unit 311 or the motion parallax compensation unit 312.

[0271]

The intra-prediction unit 311 performs basically the same process as the intra-prediction unit 114 of Fig. 11. However, the intra-prediction unit 311 performs intra-prediction only with respect to a region in which a predicted image is generated by intra-prediction during encoding.

[0272]

The motion parallax compensation unit 312 performs motion parallax compensation based on the inter-prediction information supplied from the lossless decoding unit 302 to generate a predicted image. The motion parallax compensation unit 312 performs motion parallax compensation with respect to a region in which inter-prediction is performed during encoding only based on the inter-prediction information supplied from the lossless decoding unit 302.

[0273]

The motion parallax compensation unit 312 supplies the generated predicted image to the arithmetic unit 305 via the selector 313 for every region of prediction

process units.

[0274]

The selector 313 supplies the predicted image supplied from the intra-prediction unit 311 or the predicted image supplied from the motion parallax compensation unit 312 to the arithmetic unit 305.

[0275]

Although the decoded picture buffer 309 stores the image of a processing target view (and the view ID and POC of the image) only, the decoded multi-view picture buffer 321 stores the images of respective viewpoints (views) (and the view IDs and POCs of the images). That is, the decoded multi-view picture buffer 321 acquires the decoded image (and the view ID and POC of the image) supplied to the decoded picture buffer 309 and stores the decoded image (and the view ID and POC of the image) together with the decoded picture buffer 309.

[0276]

Although the decoded picture buffer 309 erases the decoded image when a processing target view changes, the decoded multi-view picture buffer 321 stores the decoded image as it is. Moreover, the decoded multi-view picture buffer 321 supplies the stored decoded image (and the view ID and POC of the image) to the decoded picture buffer 309 as a "decoded image of a non-processing target view" according to a request of the decoded picture buffer 309 or the like. The decoded picture buffer 309 supplies the "decoded image of the non-processing target view (and the view ID and POC of the image)" read from the decoded multi-view picture buffer 321 to the motion parallax compensation unit 312 via the selector 310.

[0277]

[Motion parallax compensation unit]

Fig. 32 is a block diagram illustrating a main configuration example of the motion parallax compensation unit 312.

[0278]

As illustrated in Fig. 32, the motion parallax compensation unit 312 includes an encoded information accumulation buffer 331, a spatial correlation predictive vector generating unit 332, a temporal parallax correlation predictive vector generating unit 333, a selector 334, an arithmetic unit 335, and a predicted image generating unit 336.

[0279]

The encoded information accumulation buffer 331 acquires the mode information, difference motion parallax information, and prediction information obtained in the lossless decoding unit 302. Moreover, the encoded information accumulation buffer 331 stores the decoded motion parallax vector used in the predicted image generating unit 336. The motion parallax vector is used as a motion parallax vector of the neighboring region in the process for another region.

[0280]

The encoded information accumulation buffer 331 supplies the mode information or the decoded motion parallax vector of the neighboring region to one of the spatial correlation predictive vector generating unit 332 and the temporal parallax correlation predictive vector generating unit 333 corresponding to the type (spatial correlation predictive vector or temporal parallax

correlation predictive vector) of the predictive vector designated in the prediction information.

[0281]

Moreover, the encoded information accumulation buffer 331 supplies the difference motion parallax vector included in the difference motion parallax information to the arithmetic unit 335. Further, the encoded information accumulation buffer 331 supplies the reference image index included in the prediction information to the predicted image generating unit 336.

[0282]

The spatial correlation predictive vector generating unit 332 generates the spatial correlation predictive vector based on the information supplied from the encoded information accumulation buffer 331. The generation method is the same as that of the spatial correlation predictive vector generating unit 135. However, since the optimal inter-prediction mode is determined in advance, the spatial correlation predictive vector generating unit 332 may generate the spatial correlation predictive vector in the mode only. The spatial correlation predictive vector generating unit 332 supplies the generated spatial correlation predictive vector to the arithmetic unit 335 via the selector 334.

[0283]

The temporal parallax correlation predictive vector generating unit 333 generates a temporal parallax correlation predictive vector based on the information such as the information supplied from the encoded information accumulation buffer 331, the information such as the view ID and POC of the current region supplied

from the lossless decoding unit 302, and the information such as the view ID and POC of the reference image supplied from the decoded picture buffer 309. The generation method is the same as that of the temporal parallax correlation predictive vector generating unit 136. However, since the optimal inter-prediction mode is determined in advance, the temporal parallax correlation predictive vector generating unit 333 may generate the temporal parallax correlation predictive vector in the mode only. The temporal parallax correlation predictive vector generating unit 333 supplies the generated temporal parallax correlation predictive vector to the arithmetic unit 335 via the selector 334.

[0284]

When the spatial correlation predictive vector is supplied from the spatial correlation predictive vector generating unit 332, the selector 334 supplies the vector to the arithmetic unit 335. Moreover, the temporal parallax correlation predictive vector is supplied from the temporal parallax correlation predictive vector generating unit 333, the selector 334 supplies the vector to the arithmetic unit 335.

[0285]

The arithmetic unit 335 adds the difference motion parallax vector supplied from the encoded information accumulation buffer 331 to the spatial correlation predictive vector or the temporal parallax correlation predictive vector supplied from the selector 334 to reconstruct the motion parallax vector of the current region. The arithmetic unit 335 supplies the reconstructed motion parallax vector of the current

region to the predicted image generating unit 336.

[0286]

The predicted image generating unit 336 generates a predicted image using the reconstructed motion parallax vector of the current region supplied from the arithmetic unit 335, the reference image index supplied from the encoded information accumulation buffer 331, the pixel values of the neighboring images which are the images of the neighboring region supplied from the decoded picture buffer 309. The predicted image generating unit 336 supplies the generated predicted image pixel value to the selector 313.

[0287]

By doing so, the temporal parallax correlation predictive vector generating unit 333 can generate the parallax correlation predictive vector as well as the motion correlation predictive vector similarly to the temporal parallax correlation predictive vector generating unit 136. Thus, the motion parallax compensation unit 312 can reconstruct the parallax correlation predictive vector even when the vector of the current region is the parallax vector. That is, the image decoding device 300 can improve the encoding efficiency since the image decoding device 300 can correctly decode the encoded data generated by the image encoding device 100.

[0288]

[Flow of decoding process]

Next, the flow of the respective processes executed by the image decoding device 300 having such a configuration will be described. First, an example of

the flow of the decoding process will be described with reference to the flowchart of Fig. 33.

[0289]

When the decoding process starts, in step S301, the accumulation buffer 301 accumulates the bit stream transmitted thereto. In step S302, the lossless decoding unit 302 decodes the bit stream (encoded difference image information) supplied from the accumulation buffer 301. In this case, various types of information other than the difference image information included in the bit stream, such as intra-prediction information or inter-prediction information are also decoded.

[0290]

In step S303, the inverse quantization unit 303 performs inverse quantization on the quantized orthogonal transform coefficients obtained by the process of step S302. In step S304, the inverse orthogonal transform unit 304 performs inverse orthogonal transform on the orthogonal transform coefficients having been subjected to inverse quantization in step S303.

[0291]

In step S305, the intra-prediction unit 311 or the motion parallax compensation unit 312 performs a prediction process using the supplied information. In step S306, the arithmetic unit 305 adds the predicted image generated in step S305 to the difference image information obtained through inverse orthogonal transform in step S304. In this way, the reconstructed image is generated.

[0292]

In step S307, the loop filter 306 appropriately

performs a loop filtering process including a deblocking filtering process, an adaptive loop filtering process, and the like with respect to the reconstructed image obtained in step S306.

[0293]

In step S308, the screen rearrangement buffer 307 rearranges the decoded image generated through the filtering process in step S307. That is, the frame order rearranged for encoding by the screen rearrangement buffer 102 of the image encoding device 100 is rearranged in the original display order.

[0294]

In step S309, the D/A converter 308 performs D/A conversion on the decoded image in which the frame order is rearranged. The decoded image is output to a display (not illustrated) and is displayed.

[0295]

In step S310, the decoded picture buffer 309 stores the decoded image obtained through the filtering process in step S307. This decoded image is used as a reference image in the inter-prediction process.

[0296]

When the process of step S310 ends, the decoding process ends.

[0297]

[Flow of prediction process]

Next, an example of the flow of the prediction process executed in step S305 of Fig. 33 will be described with reference to the flowchart of Fig. 34.

[0298]

When the prediction process starts, in step S331,

the lossless decoding unit 302 determines whether the current region of a processing target has been subjected to intra-prediction during encoding. When it is determined that the current region has been subjected to intra-prediction, the lossless decoding unit 302 proceeds to the process of step S332.

[0299]

In this case, the intra-prediction unit 311 acquires intra-prediction mode information from the lossless decoding unit 302 in step S332 and generates a predicted image by intra-prediction in step S333. When the predicted image is generated, the intra-prediction unit 311 ends the prediction process and the flow returns to the flowchart of Fig. 33.

[0300]

Moreover, when it is determined in step S331 the current region is a region having been subjected to the inter-prediction, the lossless decoding unit 302 proceeds to the process of step S334. In step S334, the motion parallax compensation unit 312 performs a motion parallax compensation process. When the motion parallax compensation process ends, the motion parallax compensation unit 312 ends the prediction process and the flow returns to the flowchart of Fig. 33.

[0301]

[Flow of motion parallax compensation process]

Next, an example of the flow of the motion parallax compensation process executed in step S334 of Fig. 34 will be described with reference to the flowchart of Fig. 35.

[0302]

When the motion parallax compensation process starts, in step S351, the encoded information accumulation buffer 331 stores mode information, motion parallax information, and prediction information decoded in step S351.

[0303]

In step S352, the spatial correlation predictive vector generating unit 332, the temporal parallax correlation predictive vector generating unit 333, the selector 334, and the arithmetic unit 335 perform a motion parallax vector generation process to reconstruct a motion parallax vector of the current region.

[0304]

When the motion parallax vector is reconstructed, in step S353, the predicted image generating unit 336 generates a predicted image using the motion parallax vector.

[0305]

When the predicted image is generated, the predicted image generating unit 336 ends the motion parallax compensation process, and the flow returns to the flowchart of Fig. 34.

[0306]

[Flow of motion parallax vector generation process]

Next, an example of the flow of the motion parallax vector generation process executed in step S352 of Fig. 35 will be described with reference to the flowchart of Fig. 36.

[0307]

When the motion parallax vector generation process starts, in step S371, the encoded information

accumulation buffer 331 determines whether this mode is a skip mode from the prediction information. When it is determined that this mode is the skip mode, the encoded information accumulation buffer 331 proceeds to the process of step S372. In step S372, the spatial correlation predictive vector generating unit 332 to the arithmetic unit 335 performs a merge mode process to reconstruct the motion parallax vector in the merge mode. In the merge mode process, the same processes as the processes described with reference to the flowchart of Fig. 22 are performed. When the merge mode process ends, the arithmetic unit 335 ends the motion parallax vector generation process, and the flow returns to the flowchart of Fig. 35.

[0308]

Moreover, when it is determined in step S371 of Fig. 36 that it is not the skip mode, the encoded information accumulation buffer 331 proceeds to the process of step S373. In step S373, the encoded information accumulation buffer 331 determines whether this mode is the merge mode from the prediction information. When it is determined that this mode is the merge mode, the encoded information accumulation buffer 331 returns to the process of step S372 to execute the merge mode process.

[0309]

Moreover, when it is determined in step S373 that this mode is not the merge mode, the encoded information accumulation buffer 331 proceeds to the process of step S374.

[0310]

In step S374, the encoded information accumulation

buffer 331 acquires the index of the reference image. In step S375, the encoded information accumulation buffer 331 acquires the difference motion parallax vector.

[0311]

In step S376, the spatial correlation predictive vector generating unit 332 or the temporal parallax correlation predictive vector generating unit 333 performs a parallax motion vector prediction process. This parallax motion vector prediction process is performed in the same manner as that described with reference to the flowchart of Fig. 23. However, in this case, since the prediction method is determined in advance, any one (one designated by the prediction information) of the spatial correlation prediction process and the temporal parallax correlation prediction process is performed.

[0312]

In step S377, the arithmetic unit 335 adds the predictive vector reconstructed in step S376 and the difference motion parallax vector to reconstruct the motion parallax vector.

[0313]

When the process of step S377 ends, the arithmetic unit 335 ends the motion parallax vector generation process and the flow returns to the flowchart of Fig. 35.

[0314]

By executing the respective processes in the above-described manner, the temporal parallax correlation predictive vector generating unit 333 can generate the parallax correlation predictive vector as well as the motion correlation predictive vector similarly to the

temporal parallax correlation predictive vector generating unit 136. Thus, the motion parallax compensation unit 312 can reconstruct the parallax correlation predictive vector even when the vector of the current region is a parallax vector. That is, the image decoding device 300 can improve the encoding efficiency since the image decoding device 300 can correctly decode the encoded data generated by the image encoding device 100.

[0315]

[3. Third embodiment]

[Point]

In the case of multi-view images, the positions of images are offset between views so that a parallax occurs. Thus, when a block in the view direction is selected, even if a block at the same position is selected (referred to), the prediction accuracy of the predicted image may decrease and there is a possibility that it is not possible to create an appropriate predictive vector.

[0316]

Thus, when a block in the view direction is selected in order to generate a predictive vector, a block at a shifted position is selected. That is, the predictive vector is generated using a vector of such a region that is located at the same position as the current region in a state where the position of an image of the same time as the current region is shifted.

[0317]

The shift amount is calculated in a predetermined order from the parallax vector of the neighboring block. By using the same order in both an encoding-side device

and a decoding-side device, the same prediction can be performed in both the encoding side and the decoding side.

[0318]

A neighboring block for computing the shift amount may be explicitly designated, and the information thereof may be transmitted from the encoding side to the decoding side. Moreover, the information on a shift amount computing method may be transmitted from the encoding side to the decoding side.

[0319]

By doing so, the image encoding device and the image decoding device can generate a predictive vector from blocks aligned between views. Therefore, it is possible to improve the prediction accuracy of the predicted image and to improve the encoding efficiency.

[0320]

This will be described in detail below.

[0321]

[Image encoding device]

Fig. 37 is a block diagram illustrating another configuration example of an image encoding device to which the present technique is applied.

[0322]

An image encoding device 400 illustrated in Fig. 37 is basically the same device as the image encoding device 100 described above. However, the image encoding device 400 generates the predictive vector from blocks aligned between views.

[0323]

As illustrated in Fig. 37, the image encoding device 400 includes a motion prediction/compensation unit

415 and a base view encoder 421.

[0324]

The motion prediction/compensation unit 412 generates a predictive vector that refers to blocks in the view direction using the decoded image acquired from the decoded picture buffer 112 and the motion information of the base view acquired from the base view encoder 421.

[0325]

The base view encoder 421 encodes the base view. The base view encoder 421 supplies the decoded image of the base view to the decoded picture buffer 112 which stores the decoded image. The decoded picture buffer 112 also stores the decoded image of a non-base view supplied from the loop filter 111.

[0326]

The base view encoder 421 supplies the motion information of the base view to the motion prediction/compensation unit 412.

[0327]

[Motion prediction/compensation unit]

Fig. 38 is a block diagram illustrating a main configuration example of the motion prediction/compensation unit 412 of Fig. 37.

[0328]

As illustrated in Fig. 38, the motion prediction/compensation unit 412 includes an inter-mode generating unit 431, a reference index determining unit 432, a vector predicting unit 433, a vector predicting unit 434, and a mode determining unit 435. The information illustrated in Fig. 38 is exchanged between the respective processing units.

[0329]

In the case of the inter-prediction mode, the vector predicting unit 433 generates a predictive vector to generate the predicted image thereof. In the case of the skip mode, the merge mode, or the like, the vector predicting unit 434 generates the predictive vector to generate the predicted image thereof. The predictive vector and the predicted image generated in these vector predicting units are supplied to the mode determining unit 435.

[0330]

The mode determining unit 435 determines a mode based on these items of information and supplies mode information indicating the selected mode and the predictive vector of the mode to the lossless encoding unit 106. Moreover, the predicted image of the selected mode is supplied to the predicted image selector 116.

[0331]

[Vector predicting unit]

Fig. 39 is a block diagram illustrating a main configuration example of the vector predicting unit 433.

[0332]

As illustrated in Fig. 39, the vector predicting unit 433 includes a motion/parallax vector search unit 451, a predicted image generating unit 452, a vector cost calculating unit 453, a vector determining unit 454, a vector information accumulation buffer 455, a neighboring block-based predictive vector generating unit 456, a different picture-based predictive vector generating unit 457, and a selector 458. The information illustrated in Fig. 39 is exchanged between the respective processing

units.

[0333]

The different picture-based predictive vector generating unit 457 generates a predictive vector that refers to different pictures. That is, the different picture-based predictive vector generating unit 457 refers to different pictures in the temporal direction and the view direction to generate the predictive vector. When generating the predictive vector that refers to different pictures in the view direction, the different picture-based predictive vector generating unit 457 acquires the motion information of the base view from the base view encoder 421 and generates a predictive vector using the motion information.

[0334]

The predictive vector generated by the different picture-based predictive vector generating unit 457 is supplied to the vector cost calculating unit 453 via the selector 458, and the cost function value used for the mode determination is calculated.

[0335]

[Different picture-based predictive vector generating unit]

Fig. 40 is a block diagram illustrating a main configuration example of the different picture-based predictive vector generating unit 457.

[0336]

As illustrated in Fig. 40, the different picture-based predictive vector generating unit 457 includes a parallax vector determining unit 471, an inter-view reference vector generating unit 472, and an intra-view

reference vector generating unit 473.

[0337]

The parallax vector determining unit 471 calculates a shift amount of the reference image from the parallax vector of the neighboring block. The shift amount calculating method is optional. For example, any one of the parallax vectors of the neighboring blocks may be selected and may be used as the shift amount. Moreover, for example, an average value or a median value of the parallax vectors of the neighboring blocks may be used as the shift amount.

[0338]

The parallax vector determining unit 471 supplies the shift amount obtained in this manner to the inter-view reference vector generating unit 472 as the parallax vector.

[0339]

The inter-view reference vector generating unit 472 generates a predictive vector that refers to different pictures in the view direction.

[0340]

The inter-view reference vector generating unit 472 generates the predictive vector by taking the parallax vector into consideration using the parallax vector (shift amount) selected by the parallax vector determining unit 471, the motion vector (also including the parallax vector in the case of the non-base view) of the base view supplied from the base view encoder 421, and information such as the reference image index or the inter-view motion/parallax vector of the same time, which is read from the vector information accumulation buffer

455.

[0341]

That is, the inter-view reference vector generating unit 472 aligns (shifts) the images of the view referring to using the shift amount calculated by the parallax vector determining unit 471. Moreover, the inter-view reference vector generating unit 472 generates the predictive vector from the aligned blocks.

[0342]

The inter-view reference vector generating unit 472 supplies the generated predictive vector to the vector cost calculating unit 453 via the selector 458 (not illustrated in Fig. 40).

[0343]

The intra-view reference vector generating unit 473 generates the predictive vector that refers to different pictures in the temporal direction.

[0344]

By doing so, the inter-view reference vector generating unit 472 can create an appropriate predictive vector with high prediction accuracy. In this way, the image encoding device 400 can improve the encoding efficiency.

[0345]

[Flow of motion prediction/compensation process]

An example of the flow of the motion prediction/compensation process will be described with reference to the flowchart of Fig. 41.

[0346]

When the motion prediction/compensation process starts, in step S401, the inter-mode generating unit 431

selects any one of an inter-prediction mode, a skip mode, a merge mode, and the like and generates an inter mode which is information designating the selected mode.

[0347]

In step S402, the inter-mode generating unit 431 determines whether the generated inter mode is the inter-prediction mode.

[0348]

When the inter mode is the inter-prediction mode, the inter-mode generating unit 431 determines the reference image in step S403, and the vector predicting unit 433 executes the vector prediction process in step S404.

[0349]

Moreover, when the inter mode is not the inter-prediction mode, in step S404, the vector predicting unit 434 performs a vector prediction process.

[0350]

In step S405, the mode determining unit 435 determines the mode based on the predictive vector or the like generated in step S404. This mode is used in the process of step S401.

[0351]

In step S405, the lossless encoding unit 106 encodes the information of the mode determined in step S405.

[0352]

[Flow of vector prediction process]

An example of the flow of the vector prediction process executed by the vector predicting unit 433 in step S404 of Fig. 41 will be described with reference to

the flowchart of Fig. 42.

[0353]

When the vector prediction process starts, the motion/parallax vector search unit 451 searches vectors in step S421.

[0354]

In step S422, the predicted image generating unit 452 generates a predicted image.

[0355]

In step S423, the vector cost calculating unit 453 generates a residual image.

[0356]

In step S424, the neighboring block-based predictive vector generating unit 456 and the different picture-based predictive vector generating unit 457 generates a predictive vector from the encoded vector.

[0357]

In step S425, the vector cost calculating unit 453 calculates the residue of the vector.

[0358]

In step S426, the vector determining unit 454 determines a predictive vector having the smallest cost. This processing result is reflected on the process of step S424.

[0359]

In step S427, the vector information accumulation buffer 455 accumulates the vector information and the flow returns to the flowchart of Fig. 41.

[0360]

[Flow of predictive vector generation process]

Next, an example of the flow of the predictive

vector generation process executed in step S424 of Fig. 42 will be described with reference to the flowchart of Fig. 43.

[0361]

When the predictive vector generation process starts, in step S441, the selector 458 determines which block is to be referred to.

[0362]

When it is determined that the neighboring block of the picture is to be referred to, in step S442, the neighboring block-based predictive vector generating unit 456 sets the encoded vector of the neighboring block to the predictive vector and the flow returns to the flowchart of Fig. 42.

[0363]

Moreover, when it is determined in step S441 that the block of a different picture is to be referred to, the selector 458 proceeds to the process of step S443 to determine a view of which the picture is to be referred to.

[0364]

When it is determined that the picture of a different view is to be referred to, in step S444, the different picture-based predictive vector generating unit 457 generates a predictive vector of the view direction from a encoded co-located block that is expanded in the view direction and the flow returns to the flowchart of Fig. 42.

[0365]

Moreover, when it is determined in step S443 that the picture of the same view is to be referred to, in

step S445, the different picture-based predictive vector generating unit 457 generates the predictive vector of the temporal direction from the encoded co-located block and the flow returns to the flowchart of Fig. 42.

[0366]

[Flow of different picture-based predictive vector generation process]

Next, an example of the flow of the different picture-based predictive vector generation process executed in step S444 of Fig. 43 will be described with reference to the flowchart of Fig. 44.

[0367]

When the different picture-based predictive vector generation process starts, in step S461, the parallax vector determining unit 471 determines a shift amount from the parallax vector of the neighboring block.

[0368]

In step S462, the inter-view reference vector generating unit 472 selects a co-located block at a shifted position.

[0369]

In step S463, the inter-view reference vector generating unit 472 generates the predictive vector from the co-located block and the flow returns to the flowchart of Fig. 43.

[0370]

[Flow of shift amount determining process]

Next, an example of the flow of the shift amount determining process executed in step S461 of Fig. 44 will be described with reference to the flowchart of Fig. 45.

[0371]

In step S481, the parallax vector determining unit 471 determines whether there is a plurality of blocks in which the value of the Y-direction vector of the parallax vector of the neighboring block is not zero.

[0372]

The neighboring block is a block located near (including "adjacent") a current block (current block) of a processing target. For example, as illustrated in Fig. 46, a block (Left) adjacent to the left of a current block (Curr), a block (Above) adjacent above the current block, and a block (Above Right) adjacent to the top right corner of the current block are used as the neighboring blocks. Naturally, a block other than these blocks may be included in the neighboring block, and part or all of these blocks may not be used as the neighboring block.

[0373]

The positions of the neighboring blocks may be the same for all blocks and may be different from block to block. For example, when a block adjacent to a screen end or a slice boundary are used as the current block, part of the neighboring blocks may be set to be not usable.

[0374]

When it is determined in step S481 that there is not a plurality of blocks in which the value of the Y-direction vector is non-zero, the parallax vector determining unit 471 proceeds to the process of step S482 to use the parallax vector in the X-direction of the target as the shift amount, and the flow returns to the flowchart of Fig. 44.

[0375]

Moreover, when it is determined in step S481 that there is a plurality of blocks in which the value of the Y-direction vector is non-zero, the parallax vector determining unit 471 proceeds to the process of step S483 to set the average value of the parallax vectors in the X-direction of the target as the shift amount, and the flow returns to the flowchart of Fig. 44.

[0376]

By executing the processes in the above-described manner, the motion prediction/compensation unit 415 can create an appropriate predictive vector with high prediction accuracy. In this way, the image encoding device 400 can improve the encoding efficiency.

[0377]

[4. Fourth embodiment]

[Image decoding device]

Fig. 47 is a block diagram illustrating another configuration example of an image decoding device to which the present technique is applied.

[0378]

An image decoding device 500 illustrated in Fig. 47 is basically the same device as the image decoding device 300 described above. However, the image decoding device 500 generates a predictive vector from blocks aligned between views similarly to the image encoding device 400.

[0379]

As illustrated in Fig. 47, the image decoding device 500 includes a motion compensation unit 512 and a base view decoder 521.

[0380]

The motion compensation unit 512 generates a predictive vector that refers to blocks in the view direction using the decoded image acquired from the decoded picture buffer 309 and the motion information of the base view acquired from the base view decoder 521.

[0381]

The base view decoder 521 encodes the base view. The base view decoder 521 supplies the decoded image of the base view to the decoded picture buffer 309 which stores the decoded image. The decoded picture buffer 309 also stores the decoded image of the non-base view supplied from the loop filter 306.

[0382]

The base view decoder 521 supplies the motion information of the base view to the motion compensation unit 512.

[0383]

[Motion compensation unit]

Fig. 48 is a block diagram illustrating a main configuration example of the motion compensation unit 512 of Fig 47.

[0384]

As illustrated in Fig. 48, the motion compensation unit 512 includes a mode determining unit 531, a reference index determining unit 532, a vector decoding unit 533, and a vector decoding unit 534. The information illustrated in Fig. 48 is exchanged between the respective processing units.

[0385]

In the case of the inter-prediction mode, the vector decoding unit 533 decodes a residual vector

transmitted from the image encoding device 400 to generate a predictive vector. Moreover, the vector decoding unit 533 generates a predicted image using the predictive vector. The predicted image is supplied to the predicted image selector 313.

[0386]

In the case of the skip mode or the merge mode, the vector decoding unit 534 decodes the residual vector transmitted from the image encoding device 400 to generate the predictive vector. Moreover, the vector decoding unit 534 generates a predicted image using the predictive vector. The predicted image is supplied to the predicted image selector 313.

[0387]

[Vector decoding unit]

Fig. 49 is a block diagram illustrating a main configuration example of the vector decoding unit 533.

[0388]

As illustrated in Fig. 49, the vector decoding unit 533 includes a selector 551, a neighboring block-based predictive vector generating unit 552, a different picture-based predictive vector generating unit 553, a selector 554, an arithmetic unit 555, a predicted image generating unit 556, and a vector information accumulation buffer 557. The information illustrated in Fig. 49 is exchanged between the respective processing units.

[0389]

The selector 551 supplies the vector index supplied from the lossless decoding unit 302 to the neighboring block-based predictive vector generating unit 552 (in the

case of the skip mode, the merge mode, or the like) or to the different picture-based predictive vector generating unit 553 (in the case of the inter-prediction mode) according to the inter mode.

[0390]

The neighboring block-based predictive vector generating unit 552 supplied with the vector index generates a predictive vector from the neighboring blocks in the current picture using the vector information acquired from the vector information accumulation buffer 557.

[0391]

The different picture-based predictive vector generating unit 553 supplied with the vector index generates a predictive vector from different pictures in the current view using the vector information acquired from the vector information accumulation buffer 557. Moreover, the different picture-based predictive vector generating unit 553 generates the predictive vector from different pictures of different views using the vector information acquired from the vector information accumulation buffer 557 or the motion information of the base view supplied from the base view decoder 521.

[0392]

The different picture-based predictive vector generating unit 553 is the same processing unit as the parallax vector determining unit 471 of the different picture-based predictive vector generating unit 457 and generates the predictive vector according to the same method.

[0393]

The selector 554 supplies the predictive vector generated by the neighboring block-based predictive vector generating unit 552 or the predictive vector generated by the different picture-based predictive vector generating unit 553 to the arithmetic unit 555.

[0394]

The arithmetic unit 555 adds the difference value (residue vector) of the motion/parallax vector supplied from the lossless decoding unit 302 and the predictive vector to generate a motion/parallax vector of the current region. The arithmetic unit 555 supplies the motion/parallax vector to the predicted image generating unit 556. Moreover, the arithmetic unit 555 supplies the motion/parallax vector to the vector information accumulation buffer 557 which stores the motion/parallax vector.

[0395]

The predicted image generating unit 556 generates a predicted image using the motion/parallax vector of the current region supplied from the arithmetic unit 555, the reference image index supplied from the lossless decoding unit 302, and the decoded image pixel value supplied from the decoded picture buffer 309. The predicted image generating unit 556 supplies the generated predicted image pixel value to the predicted image selector 313.

[0396]

[Different picture-based predictive vector generating unit]

Fig. 50 is a block diagram illustrating a main configuration example of the different picture-based predictive vector generating unit 553.

[0397]

As illustrated in Fig. 50, the different picture-based predictive vector generating unit 553 includes a parallax vector determining unit 571, an inter-view reference vector generating unit 572, and an intra-view reference vector generating unit 573.

[0398]

The parallax vector determining unit 571, the inter-view reference vector generating unit 572, and the intra-view reference vector generating unit 573 are the same processing units and perform the same processes as the parallax vector determining unit 471, the inter-view reference vector generating unit 472, and the intra-view reference vector generating unit 473 of the different picture-based predictive vector generating unit 457, respectively.

[0399]

That is, the parallax vector determining unit 571 calculates a shift amount of an image of a view of a reference destination according to the same method as the parallax vector determining unit 471 and shifts the image.

[0400]

Thus, the different picture-based predictive vector generating unit 553 can correctly decode the residual vector to generate the same appropriate predictive vector with high prediction accuracy as the predictive vector generated by the different picture-based predictive vector generating unit 457. That is, the vector decoding unit 533 can generate the same predicted image as the predicted image generated by the vector predicting unit 433. Therefore, the image decoding device 500 can

improve the encoding efficiency since the image decoding device 500 can correctly decode the encoded data generated by the image encoding device 400.

[0401]

The shift amount calculating method of the parallax vector determining unit 571 is not limited as long as it is the same as the parallax vector determining unit 471 but the method is optional. For example, any one of the parallax vectors of the neighboring blocks may be selected and the selected parallax vector may be used as the shift amount. Moreover, for example, an average value or a median value of the parallax vectors of the neighboring blocks may be used as the shift amount.

[0402]

[Flow of motion compensation process]

An example of the flow of the motion compensation process will be described with reference to the flowchart of Fig. 51.

[0403]

When the motion prediction/compensation process starts, the mode determining unit 531 decodes the inter mode in step S501 and determines whether the inter mode is an inter-prediction mode in step S502.

[0404]

When the inter mode is the inter-prediction mode, the vector decoding unit 533 determines a reference image in step S503 and performs a vector decoding process to decode the residual vector to generate a predicted image in step S504.

[0405]

Moreover, when the inter mode is not the inter-

prediction mode, the vector decoding unit 534 performs a vector decoding process to decode the residual vector to generate a predicted image in step S504.

[0406]

[Flow of vector decoding process]

An example of the flow of the vector decoding process executed by the vector decoding unit 533 in step S504 of Fig. 51 will be described with reference to the flowchart of Fig. 52.

[0407]

When the vector decoding process starts, in step S521, the lossless decoding unit 302 decodes the residual vector (difference vector).

[0408]

In step S522, the lossless decoding unit 302 decodes the reference image index.

[0409]

In step S523, the lossless decoding unit 302 decodes the vector index.

[0410]

In step S524, the neighboring block-based predictive vector generating unit 552 and the different picture-based predictive vector generating unit 553 generate a predictive vector from the encoded vector. The arithmetic unit 555 adds the predictive vector to the residual vector to generate the motion/parallax vector of the current region.

[0411]

In step S525, the predicted image generating unit 556 generates the predicted image using the motion/parallax vector generated in step S524.

[0412]

In step S526, the vector information accumulation buffer 455 accumulates the vector information and the flow returns to the flowchart of Fig. 51.

[0413]

[Flow of predictive vector generation process]

Next, an example of the flow of the predictive vector generation process executed in step S524 of Fig. 52 will be described with reference to the flowchart of Fig. 53.

[0414]

When the predictive vector generation process starts, in step S541, the selector 554 determines a block that is to be referred to.

[0415]

When it is determined that a neighboring block of a current picture is to be referred to, in step S542, the neighboring block-based predictive vector generating unit 552 uses the encoded vector of the neighboring block as the predictive vector and the flow returns to the flowchart of Fig. 52.

[0416]

Moreover, when it is determined in step S541 that the block of a different picture is to be referred to, the selector 554 proceeds to the process of step S543 to determine a view of which the picture is to be referred to.

[0417]

When it is determined that the picture of a different view is to be referred to, in step S544, the different picture-based predictive vector generating unit

553 generates a predictive vector of the view direction from a encoded co-located block that is expanded in the view direction and the flow returns to the flowchart of Fig. 52.

[0418]

Moreover, when it is determined in step S543 that the picture of the same view is to be referred to, in step S545, the different picture-based predictive vector generating unit 553 generates the predictive vector of the temporal direction from the encoded co-located block and the flow returns to the flowchart of Fig. 52.

[0419]

[Flow of different picture-based predictive vector generation process]

Next, an example of the flow of the different picture-based predictive vector generation process executed in step S544 of Fig. 53 will be described with reference to the flowchart of Fig. 54.

[0420]

When the different picture-based predictive vector generation process starts, in step S561, the parallax vector determining unit 571 determines a shift amount from the parallax vector of the neighboring block.

[0421]

In step S562, the inter-view reference vector generating unit 572 selects a co-located block at a shifted position.

[0422]

In step S563, the inter-view reference vector generating unit 572 generates the predictive vector from the co-located block and the flow returns to the

flowchart of Fig. 53.

[0423]

That is, the process is performed in the same flow as the flowchart of Fig. 44.

[0424]

[Flow of shift amount determining process]

Next, an example of the flow of the shift amount determining process executed in step S561 of Fig. 54 will be described with reference to the flowchart of Fig. 55.

[0425]

In step S581, the parallax vector determining unit 571 determines whether there is a plurality of blocks in which the value of the Y-direction vector of the parallax vector of the neighboring block is non-zero.

[0426]

When it is determined in step S581 that there is not a plurality of blocks in which the value of the Y-direction vector is non-zero, the parallax vector determining unit 571 proceeds to the process of step S582 to use the parallax vector in the X-direction of the target as the shift amount, and the flow returns to the flowchart of Fig. 54.

[0427]

Moreover, when it is determined in step S581 that there is a plurality of blocks in which the value of the Y-direction vector is non-zero, the parallax vector determining unit 571 proceeds to the process of step S583 to use the average value of the parallax vectors in the X-direction of the target as the shift amount, and the flow returns to the flowchart of Fig. 54.

[0428]

That is, the process is performed in the same flow as the flowchart of Fig. 45. The same is true for the neighboring block.

[0429]

By executing the processes in the above-described manner, the motion compensation unit 512 can create an appropriate predictive vector with high prediction accuracy. In this way, the image decoding device 500 can improve the encoding efficiency.

[0430]

As described above, the information such as the neighboring block for calculating the shift amount and the shift amount computing method may be transmitted from the image encoding device 400 to the image decoding device 500.

[0431]

[5. Fifth embodiment]

[Predictive vector]

As described in the third and fourth embodiments, a block at a shifted position can be selected when a block of a view direction is selected in order to generate a predictive vector.

[0432]

Candidates for the predictive vector may be generated using the vector (co-located block) of a co-located block which is a block at the same position, belonging to the picture of a different time, of the same view as the current region, or a global parallax vector.

[0433]

For example, as illustrated in Fig. 56, any one (for example, a left-eye image) of the right and left

images of a 3D image is used as a base view and the other image (for example, a right-eye image) is used as a dependent view.

[0434]

When a vector (coding vector) 621 of a current region 611 of a current picture 601 of the dependent view is predicted, a predictive vector may be obtained using a motion vector (co-located vector) 622 of a co-located block 612 at the same position as the current region, of a co-located picture 602 which is a picture of a different time, of the same view as the current region, for example, or a global parallax vector (not illustrated).

[0435]

For example, a vector 623 of a block 613 at a position shifted by the co-located block 622 or the global parallax vector (vector 631) from the same position as the current region 11 of a picture 603 of the base view of the same time as the current picture 601 may be used as a predictive vector (PMV) of a coding vector (MV) 621. Moreover, the co-located block 622 or the global parallax vector may be used as the predictive vector of the coding vector (MV) 621.

[0436]

By using the same order in both an encoding-side device and a decoding-side device, the same prediction can be performed in both the encoding side and the decoding side.

[0437]

Information indicating which one of the co-located block and the global parallax vector will be used,

information on the co-located block or the global parallax vector, and the like may be explicitly designated and the information thereof may be transmitted from the encoding side to the decoding side. Moreover, the information on a shift amount computing method may be transmitted from the encoding side to the decoding side.

[0438]

By doing so, the image encoding device and the image decoding device can generate a predictive vector from blocks aligned similarly between views. Therefore, it is possible to improve the prediction accuracy of the predicted image and to improve the encoding efficiency.

[0439]

[Global parallax vector]

Next, the global parallax vector will be described. The global parallax vector is a representative parallax vector that is global (for each of predetermined units such as pictures, slices, LCUs, or CUs, for example). For example, the global parallax vector generated for each picture indicates a parallax amount between views. A method of generating the global parallax vector is optional.

[0440]

A specific example of the global parallax vector is disclosed in Junghak Nam, Hyomin Choi, Sunmi Yoo, Woong Lim, Donggyu Sim, "3D-HEVC-CE3 result on KWU's advanced motion and disparity prediction method based on global disparity," INTERNATIONAL ORGANISATION FOR STANDARDISATION ORGANISATION INTERNATIONALE DE NORMALISATION ISO/IEC JTC1/SC29/WG11 CODING OF MOVING PICTURES AND AUDIO, ISO/IEC JTC1/SC29/WG11

MPEG2011/M23620, February 2012, San Jose, CA.

[0441]

Fig. 57 is a diagram for describing a parallax and a depth.

[0442]

As illustrated in Fig. 57, when a color image of a subject M is captured by a camera c1 disposed at a position C1 and a camera c2 disposed at a position C2, a depth Z which is a distance of the subject M in the depth direction from the camera c1 (camera c2) is defined by Expression (1) below.

[0443]

$$Z = (L/d) \times f \cdots (1)$$

[0444]

"L" is the distance (hereinafter, referred to as an inter-camera distance) in the horizontal direction between the positions C1 and C2. Moreover, "d" is a value (that is, a parallax (disparity)) obtained by subtracting a distance u2 in the horizontal direction from the center of the color image, of the position of the subject M on the color image captured by the camera c2 from a distance u1 in the horizontal direction from the center of the color image, of the position of the subject M on the color image captured by the camera c1. Further, "f" is a focal distance of the camera c1, and in Expression (1), it is assumed that the focal distances of the cameras c1 and c2 are the same.

[0445]

That is, the parallax d is defined by Expression (2) below.

[0446]

[Mathematical formula 1]

$$d = \frac{f}{z} \cdot L$$

... (2)

[0447]

As illustrated in Expression (1) or (2), the parallax d and the depth Z can be converted uniquely. Thus, in the present specification, an image indicating the parallax d of the 2-view color image captured by the cameras c1 and c2 and an image indicating the depth Z will be collectively referred to as a depth image (parallax image).

[0448]

The depth image (parallax image) may be an image that indicates the parallax d or the depth Z. A normalized value of the parallax d, a normalized value of a reciprocal 1/Z of the depth Z, or the like rather than the parallax d or the depth Z itself may be employed as the pixel value of the depth image (parallax image).

[0449]

The depth Z can be obtained by Expression (3) below.

[0450]

[Mathematical formula 2]

$$z = \frac{1}{\frac{d}{255} \cdot \left(\frac{1}{z_{near}} - \frac{1}{z_{far}} \right) + \frac{1}{z_{far}}}$$

... (3)

[0451]

In Expression (3), Z_{far} is the largest value of the depth Z, and Z_{near} is the smallest value of the depth Z. The largest value Z_{far} and the smallest value Z_{near} may be

set for one screen unit or may be set for a plurality of screen units.

[0452]

As described above, in the present specification, by taking the fact that the parallax d and the depth Z can be converted uniquely into consideration, an image of which the pixel value is a normalized value I of the parallax d and an image of which the pixel value is a normalized value y of the reciprocal 1/Z of the depth Z will be collectively referred to a depth image (parallax image). In this example, although a color format of the depth image (parallax image) is YUV420 or YUV400, another color format may be used.

[0453]

When the information of the value I or y itself rather than the pixel value of the depth image (parallax image) is focused on, the value I or y is used as depth information (parallax information). Further, one that maps the value I or y is used as a depth map (parallax map).

[0454]

[Generation of predictive vector]

The predictive vector is generated according to a method corresponding to properties of both a coding vector and a co-located vector, for example, as in the table illustrated in Fig. 58.

[0455]

For example, as illustrated on the sixth row from the bottom of the table of Fig. 58, when both the coding vector and the co-located vector are motion vectors, the co-located vector is used as a predictive vector

candidate.

[0456]

Moreover, for example, as illustrated on the second row from the bottom of the table of Fig. 58, when both the coding vector and the co-located vector are parallax vectors (inter-view vectors), the co-located vector is used as a predictive vector candidate.

[0457]

In contrast, for example, as illustrated on the fifth row from the bottom of the table of Fig. 58, when the coding vector is a motion vector and the co-located vector is a parallax vector (inter-view vector), a motion vector of a base view of a block shifted by a co-located vector is used as a predictive vector candidate.

[0458]

Moreover, for example, as illustrated on the fourth row from the bottom of the table of Fig. 58, when the coding vector is a motion vector and the co-located block is an intra mode, a motion vector of a base view of a block shifted by a global parallax vector is used as a predictive vector candidate.

[0459]

Further, for example, as illustrated on the third row from the bottom of the table of Fig. 58, when the coding vector is a parallax vector (inter-view vector) and the co-located vector is a motion vector, the global parallax vector is used as a predictive vector candidate.

[0460]

Further, for example, as illustrated on the first row from the bottom of the table of Fig. 58, when the coding vector is a parallax vector (inter-view vector)

and the co-located block is an intra mode, the global parallax vector is used as a predictive vector candidate.

[0461]

By increasing the number of methods for generating the candidates for predictive vectors, it is possible to improve the prediction accuracy of the predictive vector and to improve the encoding efficiency.

[0462]

[Image encoding device]

In this case, the configuration example of the image encoding device is the same as that of the image encoding device 400 illustrated in Figs. 35 to 38, and the description thereof will not be provided.

[0463]

[Flow of process]

Moreover, the motion prediction/compensation process of this case is performed in the same manner as that described with reference to the flowchart of Fig. 39. Moreover, the vector prediction process of this case is performed in the same manner as that described with reference to the flowchart of Fig. 40. Thus, the description of these processes will not be provided.

[0464]

An example of the flow of the predictive vector generation process of this case will be described with reference to the flowchart of Fig. 59.

[0465]

When the predictive vector generation process starts, in step S601, the selector 458 determines which block is to be referred to.

[0466]

When it is determined that the neighboring block is to be referred to, the flow proceeds to step S602. In step S602, the neighboring block-based predictive vector generating unit 456 uses an encoded vector of the neighboring block as a predictive vector. When the process of step S602 ends, the predictive vector generation process ends, and the flow returns to the flowchart of Fig. 40.

[0467]

Moreover, when it is determined in step S601 that a block of a different picture is selected as a reference destination, the flow proceeds to step S603. In step S603, the different picture-based predictive vector generating unit 457 uses an encoded vector of a block of a different time/view as a predictive vector. When the process of step S603 ends, the predictive vector generation process ends and the flow returns to the flowchart of Fig. 40.

[0468]

Next, an example of the flow of the different picture-based predictive vector generation process executed in step S603 of Fig. 59 will be described with reference to the flowcharts of Figs 60 and 61.

[0469]

When the different picture-based predictive vector generation process starts, in step S621, the parallax vector determining unit 471 determines whether the coding vector is a motion vector. When it is determined that the coding vector is a motion vector, the flow proceeds to step S622.

[0470]

In step S622, the parallax vector determining unit 471 determines the mode of the co-located block. When it is determined that the mode of the co-located block is a motion vector, the flow proceeds to step S623.

[0471]

In step S623, the intra-view reference vector generating unit 473 sets the co-located vector as the predictive vector. When the process of step S623 ends, the different picture-based predictive vector generation process ends, and the flow returns to the flowchart of Fig. 59.

[0472]

Moreover, when it is determined in step S622 that the mode of the co-located block is an intra-prediction, the flow proceeds to step S624.

[0473]

In step S624, the inter-view reference vector generating unit 472 obtains a global parallax vector and sets a vector of a block of a base view shifted by the global parallax vector as a predictive vector. When the process of step S624 ends, the different picture-based predictive vector generation process ends and the flow returns to the flowchart of Fig. 59.

[0474]

Moreover, when it is determined in step S622 that the mode of the co-located block is a parallax vector, the flow proceeds to step S625.

[0475]

In step S625, the inter-view reference vector generating unit 472 sets a vector of a block of a base view shifted by the co-located vector as a predictive

vector. When the process of step S625 ends, the different picture-based predictive vector generation process ends, and the flow returns to the flowchart of Fig. 59.

[0476]

Moreover, when it is determined in step S621 that the coding vector is a parallax vector, the flow proceeds to the flowchart of Fig. 61.

[0477]

In step S631 of Fig. 61, the parallax vector determining unit 471 determines the mode of the co-located block. When it is determined that the mode of the co-located block is a parallax vector, the flow proceeds to step S632.

[0478]

In step S632, the intra-view reference vector generating unit 473 sets the co-located vector as the predictive vector. When the process of step S632 ends, the different picture-based predictive vector generation process ends, and the flow returns to the flowchart of Fig. 59.

[0479]

Moreover, when it is determined in step S631 that the mode of the co-located block is a motion vector, the flow proceeds to step S634. In step S634, the inter-view reference vector generating unit 472 sets the global parallax vector as the predictive vector. When the process of step S634 ends, the different picture-based predictive vector generation process ends, and the flow returns to the flowchart of Fig. 59.

[0480]

Further, when it is determined in step S631 that the mode of the co-located block is an intra-prediction, the flow proceeds to step S635. In step S635, the inter-view reference vector generating unit 472 sets the global parallax vector as the predictive vector. When the process of step S635 ends, the different picture-based predictive vector generation process ends, and the flow returns to the flowchart of Fig. 59.

[0481]

By executing the respective processes in the above-described manner, the image encoding device of this case can generate a predictive vector using the co-located block and the global parallax vector. Due to this, the image encoding device can improve the prediction accuracy of the predictive vector and improve the encoding efficiency.

[0482]

[Image decoding device]

Next, an image decoding device corresponding to the image encoding device of this case will be described. A configuration example of the image decoding device of this case is the same as that of the image decoding device 500 illustrated in Figs. 45 to 48 similarly to the case of the image encoding device, and the description thereof will not be provided.

[0483]

[Flow of process]

Next, the flow of various processes executed by the image decoding device of this case will be described. In this case, the image decoding device performs basically the same process as the image encoding device. That is,

the motion compensation process is executed in the same manner as that described with reference to the flowchart of Fig. 49. Moreover, the vector decoding process is executed in the same manner as that described with reference to the flowchart of Fig. 50.

[0484]

The predictive vector generation process is executed in the same manner as that (that is, that executed by the image encoding device) described with reference to the flowchart of Fig. 59.

[0485]

However, in this case, the selector 551 performs the process of step S601. Moreover, in step S602, the neighboring block-based predictive vector generating unit 552 uses a decoded vector of the neighboring block as a predictive vector. Further, in step S603, the different picture-based predictive vector generating unit 553 uses a decoded vector of a block of a different time/view as the predictive vector.

[0486]

Moreover, the different picture-based predictive vector generation process is executed in the same manner as that (that is, that executed by the image encoding device) described with reference to the flowcharts of Figs. 60 and 61.

[0487]

However, in this case, the parallax vector determining unit 571 performs the processes of steps S621, S622, and S631, the intra-view reference vector generating unit 573 performs the processes of steps S623 and S632, and the inter-view reference vector generating

unit 572 executes the processes of steps S624, S625, S634, and S635.

[0488]

By executing the respective processes in this manner, the image decoding device of this case can generate the predictive vector using the co-located block and the global parallax vector similarly to the image encoding device. Due to this, the image decoding device can improve the prediction accuracy of the predictive vector and improve the encoding efficiency.

[0489]

[6. Sixth embodiment]

[Type of reference image]

For example, in the case of HEVC, a reference image has two types which are a short reference picture and a long reference picture. Since the short reference picture is a picture that is temporally near the encoding picture, a scaling process based on the temporal distance is performed on the predictive vector. In contrast, since the long reference picture is a picture that is temporally distant from the encoding picture, scaling on the predictive vector does not create meaning, and thus, is not performed.

[0490]

By using these reference pictures appropriately according to a motion of an image or the like, it is possible to further decrease the coding rate.

[0491]

Although the types (the short reference picture or the long reference picture) of the reference pictures of the coding vector and the predictive vector are different,

the predictive vector is set to be available.

[0492]

When the types of the reference pictures of the coding vector and the predictive vector are different, it is expected that the correlation of these vectors is low. Thus, there is a possibility that the encoding efficiency decreases if the vectors are included in a vector candidate list. For example, there is a possibility that a vector having higher correlation is relegated toward the rear of the candidate list and the coding rate of the index for designating the vector increases. Moreover, for example, there is a possibility that a vector having higher correlation is relegated toward the rear of the candidate list, excluded from the candidate list, and cannot be designated.

[0493]

Thus, when the types (the short reference picture or the long reference picture) of the reference picture of the coding vector and the predictive vector are different, the predictive vector is set to be not available.

[0494]

Since the property of a vector that refers to the short reference picture is different from the property of a vector that refers to the long reference picture, it is expected that correlation is low. Thus, by excluding the vectors from the candidate vector in this case, it is possible to improve the encoding efficiency.

[0495]

For example, Fig. 62 is a diagram for describing an example of the aspect of a reference image of a fixed

background application. In the case of such a moving image (fixed background application) that a moving object is present on the front side of a background image of a still image, since the background (particularly, an occlusion region) which is a fixed region refers to a long reference picture, the motion vector tends to be 0. In contrast, since the moving object on the front side refers to a short reference picture, the motion vector occurs. If the types of reference pictures are different in this manner, the correlation of motion vectors is low (A of Fig. 64). Therefore, as described above, when the types of the reference pictures of the coding vector and the predictive vector are different, by excluding the vectors from the candidate vector, it is possible to improve the encoding efficiency.

[0496]

Moreover, for example, Fig. 63 is a diagram for describing an example of the aspect of a reference image of a stereo application. In the case of such a moving image (stereo application) that an image (left-eye image) for the left eye and an image (right-eye image) for the right eye are prepared for stereoscopic view, when the dependent view (in this example, the right-eye image) is encoded, the base view (in this example, the left-eye image) is designated as a long reference picture, and an encoded picture of the dependent view is designated as a short reference picture.

[0497]

When the dependent view is referred to, scaling can be performed since the frame IDs of a reference destination and a reference source are different.

However, when the base view is referred to, the frame IDs of the reference destination and the reference source are identical. Thus, during the scaling, the denominator may become 0, and the scaling is difficult. Therefore, in HEVC or the like, the base view is designated as a long reference picture in which the scaling is not performed.

[0498]

Therefore, in the case of the stereo application, a vector that refers to the long reference picture becomes a parallax vector and a vector that refers to the short reference picture becomes a motion vector. Thus, the correlation of motion (parallax) vectors is low between picture types (B of Fig. 64). Therefore, as described above, when the types of the reference pictures of the coding vector and the predictive vector are different, by excluding the vectors from the candidate vector, the encoding efficiency can be improved.

[0499]

In the following description, the block (CU, PU, or the like) of a processing target is referred to as an encoding block (or current) block. Moreover, a block that is temporally near the encoding block (that is, a co-located block of a picture that is temporally near a picture (current picture) in which the encoding block is present) is referred to as a temporal correlation block. Further, a block that is spatially near the encoding block (that is, a block that is adjacent to the encoding block in a current picture or a block positioned near the encoding block) is referred to as a neighboring block.

[0500]

Fig. 65 illustrates an example of a neighboring

block. The blocks at positions A0 and A1 are neighboring blocks of the encoding block (current block) and are also referred to as left blocks. Moreover, the blocks at positions B0, B1, and B2 are neighboring blocks of the encoding block (current block) and are also referred to as upper blocks.

[0501]

[Summary of handling of motion (parallax) vector]

Fig. 66 is a diagram for describing an example of handling of a temporal correlation block and a neighboring block. In vector prediction, whether a motion (parallax) vector of a temporal correlation block will be included in a candidate vector and whether scaling will be performed are determined as in the table illustrated in A of Fig. 66.

[0502]

That is, for example, when the type of a reference image of the encoding block is identical to the type of a reference image of a temporal correlation block (that is, when both reference images are short reference images or long reference images), a motion (parallax) vector of the temporal correlation block is used as a candidate. When the types of both reference images are not identical, the vectors are excluded from the candidate. Further, when both the reference image of the encoding block and the reference image of the temporal correlation block are short reference images, scaling of the motion (parallax) vector of the temporal correlation block is performed. When both reference images are long reference images, scaling of the motion (parallax) vector of the temporal correlation block is not performed.

[0503]

Moreover, in vector prediction, whether a motion (parallax) vector of a neighboring block will be included in a candidate vector and whether scaling will be performed are determined as in the table illustrated in B of Fig. 66. That is, this is the same as the case of the temporal correlation block.

[0504]

That is, for example, when the type of the reference image of the encoding block is identical to the type of the reference image of the neighboring block (that is, when both reference images are short reference images or long reference images), the motion (parallax) vector of the neighboring block is used as a candidate. When the types of both reference images are not identical, the vectors are excluded from the candidate. Further, when both the reference image of the encoding block and the reference image of the neighboring block are short reference images, scaling of the motion (parallax) vector of the neighboring block is performed. When both reference images are long reference images, scaling of the motion (parallax) vector of the neighboring block is not performed.

[0505]

[Flow of process during encoding]

An example of the flow of the process during encoding for realizing such control will be described below. Such control can be realized by the image encoding device 400 (Fig. 37) described in the third embodiment.

[0506]

The encoding process of the image encoding device 400 is performed in the same manner as that (first embodiment) described with reference to the flowchart of Fig. 20.

[0507]

An example of the flow of a PU motion (parallax) vector and reference index generation process executed by the motion prediction/compensation unit 415 as the process corresponding to the inter-motion prediction process executed in step S104 of Fig. 20 will be described with reference to the flowchart of Fig. 67.

[0508]

In step S701, the inter-mode generating unit 431 (Fig. 38) generates an inter mode. In step S702, the inter-mode generating unit 431 determines whether the inter mode is a merge (skip) mode.

[0509]

When it is determined that the inter mode is the merge (skip) mode, the flow proceeds to step S703. In step S703, the reference index determining unit 432 and the vector predicting unit 434 performs the process of the merge (skip) mode to generate a motion (parallax) vector and a reference index. When the process of step S703 ends, the flow proceeds to step S707.

[0510]

Moreover, when it is determined in step S702 that the inter mode is not the merge (skip) mode, the process proceeds to step S704. In step S704, the vector predicting unit 433 acquires a residual motion (parallax) vector and a reference index. In step S705, the vector predicting unit 433 performs the process of the AMVP mode

to generate a predictive motion (parallax) vector. In step S706, the mode determining unit 435 adds the residual motion (parallax) vector and the predictive motion (parallax) vector.

[0511]

In step S707, the mode determining unit 435 returns to step S701 before all modes are processed and determines an optimal mode when all modes have been processed.

[0512]

In step S708, the lossless encoding unit 106 encodes the selected information. When the process of step S708 ends, the flow returns to the flowchart of Fig. 20.

[0513]

Next, an example of the flow of the merge (skip) mode process executed in step S703 of Fig. 67 will be described with reference to the flowchart of Fig. 68.

[0514]

When the process starts, in step S711, the reference index determining unit 432 creates a candidate motion (parallax) vector and a reference index from spatially neighboring blocks.

[0515]

In step S712, the reference index determining unit 432 generates a reference index for temporal correlation blocks.

[0516]

In step S713, the vector predicting unit 434 generates a candidate motion (parallax) vector from temporal correlation blocks.

[0517]

In step S714, the vector predicting unit 434 generates a candidate list of motion (parallax) vectors and reference indexes. The number of elements of this list is referred to as Y.

[0518]

In step S715, the vector predicting unit 434 sets a largest number X of the candidate list.

[0519]

In step S716, the vector predicting unit 434 determines whether the number Y of elements of the list is smaller than the largest number X of the candidate list (Y < X). When it is determined that the number Y of elements of the list is smaller than the largest number X of the candidate list (Y < X), the flow proceeds to step S717.

[0520]

In step S717, the vector predicting unit 434 combines the respective elements of the candidate list to generate a new motion (parallax) vector and a new reference index.

[0521]

In step S718, the vector predicting unit 434 updates the candidate list. The number of elements of the list in this case is referred to as Y'.

[0522]

In step S719, the vector predicting unit 434 determines whether the number Y' of elements of the list is smaller than the largest number X of the candidate list (Y' < X). When it is determined that the number Y' of elements of the list is smaller than the largest

number X of the candidate list (Y' < X), the flow proceeds to step S720.

[0523]

In step S720, the vector predicting unit 434 generates a new zero motion (parallax) vector and a new zero reference index. When the process of step S720 ends, the flow proceeds to step S721.

[0524]

Moreover, when it is determined in step S716 that the number Y of elements of the list is greater than the largest number X of the candidate list (not Y < X), the flow proceeds to step S721. Moreover, when it is determined in step S719 that the number Y' of elements of the list is greater than the largest number X of the candidate list (not Y' < X), the flow proceeds to step S721.

[0525]

In step S721, the vector predicting unit 434 generates an element index of the candidate list.

[0526]

In step S722, the vector predicting unit 434 acquires a motion (parallax) vector and a reference index indicated by the element index. When the process of step S722 ends, the flow returns to the flowchart of Fig. 67.

[0527]

Next, an example of the flow of a process of generating candidate motion (parallax) vectors from a temporal correlation block executed in step S713 of Fig. 68 will be described with reference to the flowchart of Fig 69.

[0528]

When the process starts, in step S731, the vector predicting unit 434 generates an index that designates a temporal correlation picture.

[0529]

In step S732, the vector predicting unit 434 determines a temporal correlation picture.

[0530]

In step S733, the vector predicting unit 434 selects a bottom-right block of an encoding PU (encoding block) present in the temporal correlation picture.

[0531]

In step S734, the vector predicting unit 434 determines whether the bottom-right block is an intra mode or not available. When it is determined that the bottom-right block is an intra mode or not available, the flow proceeds to step S735.

[0532]

In step S735, the vector predicting unit 434 selects a central block of an encoding PU present in the temporal correlation picture.

[0533]

In step S736, the vector predicting unit 434 determines whether the central block is an intra mode or not available. When it is determined that the central block is an intra mode or not available, the flow proceeds to step S737.

[0534]

In step S737, the vector predicting unit 434 excludes the motion (parallax) vector of the temporal correlation block from the candidate. When the process of step S737 ends, the flow returns to the flowchart of

Fig. 68.

[0535]

Moreover, when it is determined in step S734 that the bottom-right block is neither an intra mode or not available, the flow proceeds to step S738. Similarly, when it is determined that the central block is neither an intra mode or not available, the flow proceeds to step S738.

[0536]

In step S738, the vector predicting unit 434 determines a motion (parallax) vector and a reference index of the temporal correlation block.

[0537]

In step S739, the vector predicting unit 434 determines the presence of the scaling process for the motion (parallax) vector of the temporal correlation block and the presence of a candidate.

[0538]

In step S740, the vector predicting unit 434 determines whether the motion (parallax) vector of the temporal correlation block is to be excluded from the candidate based on the determination result of step S739.

[0539]

When it is determined that the motion (parallax) vector is to be excluded from the candidate, the flow returns to step S737. Moreover, when it is determined in step S740 that the motion (parallax) vector is not to be excluded from the candidate (to be included in the candidate), the flow proceeds to step S741.

[0540]

In step S741, the vector predicting unit 434

determines whether scaling is necessary for the motion (parallax) vector of the temporal correlation block based on the determination result of step S739.

[0541]

When it is determined that scaling is necessary, the flow proceeds to step S742. In step S742, the vector predicting unit 434 performs a scaling process for the motion (parallax) vector of the temporal correlation block. When the process of step S742 ends, the flow returns to the flowchart of Fig. 68.

[0542]

Moreover, when it is determined in step S741 that scaling is not necessary, the flow returns to the flowchart of Fig. 68.

[0543]

Next, an example of the flow of a process of determining the presence of a scaling process for the motion (parallax) vector of the temporal correlation block and the presence of the candidate, executed in step S739 of Fig. 69 will be described with reference to the flowchart of Fig. 70.

[0544]

When the process starts, in step S751, the vector predicting unit 434 determines whether the reference image of the encoding block is a short reference image or a long reference image.

[0545]

In step S752, the vector predicting unit 434 determines whether the reference image of the temporal correlation block is a short reference image or a long reference image.

[0546]

In step S753, the vector predicting unit 434 determines whether the reference image of the encoding block is a long reference image based on the determination result of step S751.

[0547]

When it is determined that the reference image of the encoding block is a long reference image, the flow proceeds to step S754. In step S754, the vector predicting unit 434 determines whether the reference image of the temporal correlation block is a long reference image based on the determination result of step S752.

[0548]

When it is determined that the reference image of the temporal correlation block is a long reference image, the flow proceeds to step S755. In this case, the motion (parallax) vectors of the encoding block and the temporal correlation block are long reference images. Thus, in step S755, the vector predicting unit 434 includes the motion (parallax) vector of the temporal correlation block into the candidate and sets the scaling to be not necessary. When the process of step S755 ends, the flow returns to the flowchart of Fig. 69.

[0549]

Moreover, when it is determined in step S754 of Fig. 70 that the reference image of the temporal correlation block is a short reference image, the flow proceeds to step S756. In this case, the types of the motion (parallax) vectors of the encoding block and the temporal correlation block are not identical. Thus, in step S756,

the vector predicting unit 434 sets the motion (parallax) vector of the temporal correlation block to be excluded from the candidate. When the process of step S756 ends, the flow returns to the flowchart of Fig. 69.

[0550]

Further, when it is determined in step S753 of Fig. 70 that the reference image of the encoding block is a short reference image, the flow proceeds to step S757. In step S757, the vector predicting unit 434 determines whether the reference image of the temporal correlation block is a long reference image based on the determination result of step S752.

[0551]

When it is determined that the reference image of the temporal correlation block is a long reference image, the flow returns to step S756. That is, in this case, when the types of the motion (parallax) vectors of the encoding block and the temporal correlation block are not identical, the motion (parallax) vector of the temporal correlation block is set to be excluded from the candidate.

[0552]

Moreover, when it is determined in step S757 that the reference image of the temporal correlation block is a short reference image, the flow proceeds to step S758. In this case, the motion (parallax) vectors of the encoding block and the temporal correlation block are short reference images. Thus, in step S758, the vector predicting unit 434 includes the motion (parallax) vector of the temporal correlation block into the candidate and sets the scaling to be necessary. When the process of

step S758 ends, the flow returns to the flowchart of Fig. 69.

[0553]

Next, an example of the flow of the AMVP mode process executed in step S705 of Fig. 67 will be described with reference to the flowchart of Fig. 71.

[0554]

When the process starts, in step S761, the vector predicting unit 433 (Fig. 38) generates a candidate motion (parallax) vector from a spatially neighboring block.

[0555]

In step S762, the vector predicting unit 433 generates a candidate list of motion (parallax) vectors. The number of elements of this candidate list is referred to as A.

[0556]

In step S763, the vector predicting unit 433 determines whether the number A of elements of the candidate list is smaller than 2 ($A < 2$).

[0557]

When it is determined that the number A of elements of the candidate list is smaller than 2 ($A < 2$), the flow proceeds to step S764. In step S764, the vector predicting unit 433 generates a candidate motion (parallax) vector from the temporal correlation block. This process is as the same as that described with reference to the flowchart of Fig. 69, and the description thereof will not be provided.

[0558]

In step S765, the vector predicting unit 433

generates a candidate list of motion (parallax) vectors and reference indexes. The number of elements of this candidate list is referred to as A'.

[0559]

In step S766, the vector predicting unit 433 determines whether the number A' of elements of the candidate list is smaller than 2 (A' < 2).

[0560]

When it is determined that the number A' of elements of the candidate list is smaller than 2 (A' < 2), the flow proceeds to step S767. In step S767, the vector predicting unit 433 generates a new zero motion (parallax) vector and a new zero reference index. When the process of step S767 ends, the flow proceeds to step S768.

[0561]

Moreover, when it is determined in step S763 that the number A of elements of the candidate list is greater than 2 (not A < 2), the flow proceeds to step S768. Further, when it is determined in step S766 that the number A' of elements of the candidate list is greater than 2 (not A < 2), the flow proceeds to step S768.

[0562]

In step S768, the vector predicting unit 433 generates an element index (flag) of the candidate list.

[0563]

In step S769, the vector predicting unit 433 acquires a motion (parallax) vector indicated by the element index. When the process of step S769 ends, the flow returns to the flowchart of Fig. 67.

[0564]

Next, an example of the flow of a process of generating a candidate motion (parallax) vector from a spatially neighboring block executed in step S761 of Fig. 71 will be described with reference to the flowchart of Fig. 72.

[0565]

When the process starts, the vector predicting unit 433 generates a candidate motion (parallax) vector from the left block in step S771.

[0566]

In step S772, the vector predicting unit 433 generates a candidate motion (parallax) vector from the upper block.

[0567]

When the process of step S722 ends, the flow returns to the flowchart of Fig. 71.

[0568]

Next, an example of the flow of a process of generating a candidate motion (parallax) vector from the left block executed in step S771 of Fig. 72 will be described with reference to the flowcharts of Figs. 73 and 74.

[0569]

When the process starts, the vector predicting unit 433 selects the block at the position A0 in step S781.

[0570]

In step S782, the vector predicting unit 433 determines whether the block at the position A0 is an intra mode or not available.

[0571]

When it is determined that the block at the

position A0 is neither an intra mode or not available, the flow proceeds to step S783. In step S783, the vector predicting unit 433 determines whether the block at the position A0 refers to the same reference image as the encoding block.

[0572]

When it is determined that the block at the position A0 refers to the same reference image as the encoding block, the flow proceeds to step S784. In step S784, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position A0 as the candidate. When the process of step S784 ends, the flow returns to the flowchart of Fig. 72.

[0573]

Moreover, when it is determined in step S782 of Fig. 73 that the block at the position A0 is an intra mode or not available, the flow proceeds to step S785. Moreover, when it is determined in step S783 that the block at the position A0 refers to a reference image different from that of the encoding block, the flow proceeds to step S785.

[0574]

In step S785, the vector predicting unit 433 selects a block at the position A1.

[0575]

In step S786, the vector predicting unit 433 determines whether the block at the position A1 is an intra mode or not available.

[0576]

When it is determined that the block at the position A1 is an intra mode or not available, the flow

proceeds to step S787. In step S787, the vector predicting unit 433 determines whether the block at the position A1 refers to the same reference image as the encoding block.

[0577]

When it is determined that the block at the position A1 refers to the same reference image as the encoding block, the flow proceeds to step S788. In step S788, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position A1 as the candidate. When the process of step S788 ends, the flow returns to the flowchart of Fig. 72.

[0578]

Moreover, when it is determined in step S786 of Fig. 73 that the block at the position A1 is an intra mode or not available, the flow proceeds to step S791 of Fig. 74. Moreover, when it is determined in step S787 of Fig. 73 that the block at the position A1 refers to a reference image different from that of the encoding block, the flow proceeds to step S791 of Fig. 74.

[0579]

In step S791 of Fig. 74, the vector predicting unit 433 selects a block at the position A0.

[0580]

In step S792, the vector predicting unit 433 determines whether the block at the position A0 is an intra mode or not available.

[0581]

When it is determined that the block at the position A0 is neither an intra mode or not available, the flow proceeds to step S793. In step S793, the vector

predicting unit 433 uses the motion (parallax) vector of the block at the position A0 as the candidate. When the process of step S793 ends, the flow proceeds to step S797.

[0582]

Moreover, when it is determined in step S792 of Fig. 74 that the block at the position A0 is an intra mode or not available, the flow proceeds to step S794.

[0583]

In step S794, the vector predicting unit 433 selects a block at the position A1.

[0584]

In step S795, the vector predicting unit 433 determines whether the block at the position A1 is an intra mode or not available.

[0585]

When it is determined that the block at the position A0 is an intra mode or not available, the flow returns to the flowchart of Fig. 72.

[0586]

Moreover, when it is determined in step S795 of Fig. 74 that the block at the position A1 is neither an intra mode or not available, the flow returns to step S796. In step S796, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position A1 as the candidate. When the process of step S796 ends, the flow proceeds to step S797.

[0587]

In step S797, the vector predicting unit 433 determines the presence of a scaling process for the motion (parallax) vector of the neighboring block and the presence of the candidate.

[0588]

In step S798, the vector predicting unit 433 determines whether the motion (parallax) vector of the neighboring block is to be excluded from the candidate based on the determination result of step S797.

[0589]

When it is determined that the motion (parallax) vector is to be excluded from the candidate, the flow proceeds to step S799. In step S799, the vector predicting unit 433 excludes the motion (parallax) vector of the left block from the candidate. When the process of step S799 ends, the flow returns to the flowchart of Fig. 72.

[0590]

Moreover, when it is determined in step S798 of Fig. 74 that the motion (parallax) vector is not to be excluded from the candidate (to be included in the candidate), the flow proceeds to step S800.

[0591]

In step S800, the vector predicting unit 433 determines whether scaling is necessary for the motion (parallax) vector of the neighboring block based on the determination result of step S797.

[0592]

When it is determined that scaling is necessary, the flow proceeds to step S801. In step S801, the vector predicting unit 433 performs a scaling process for the motion (parallax) vector of the neighboring block. When the process of step S801 ends, the flow returns to the flowchart of Fig. 72.

[0593]

Moreover, when it is determined in step S800 of Fig. 74 that scaling is not necessary, the flow returns to the flowchart of Fig. 72.

[0594]

Next, an example of the flow of a process of determining the presence of a scaling process for the motion (parallax) vector of the neighboring block and the presence of the candidate, executed in step S797 of Fig. 74 will be described with reference to the flowchart of Fig. 75.

[0595]

When the process starts, in step S811, the vector predicting unit 433 determines whether the reference image of the encoding block is a short reference image or a long reference image.

[0596]

In step S812, the vector predicting unit 433 determines whether the reference image of the neighboring block is a short reference image or a long reference image.

[0597]

In step S813, the vector predicting unit 433 determines whether the reference image of the encoding block is a long reference image based on the determination result of step S811.

[0598]

When it is determined that the reference image of the encoding block is a long reference image, the flow proceeds to step S814. In step S814, the vector predicting unit 433 determines whether the reference image of the neighboring block is a long reference image

based on the determination result of step S812.

[0599]

When it is determined that the reference image of the neighboring block is a long reference image, the flow proceeds to step S815. In this case, the motion (parallax) vectors of the encoding block and the neighboring block are long reference images. Thus, in step S815, the vector predicting unit 433 includes the motion (parallax) vector of the neighboring block into the candidate and sets the scaling to be not necessary. When the process of step S815 ends, the flow returns to the flowchart of Fig. 74.

[0600]

Moreover, when it is determined in step S814 of Fig. 75 that the reference image of the neighboring block is a short reference image, the flow proceeds to step S816. In this case, the types of the motion (parallax) vectors of the encoding block and the neighboring block are not identical. Thus, in step S816, the vector predicting unit 433 sets the motion (parallax) vector of the neighboring block to be excluded from the candidate. When the process of step S816 ends, the flow returns to the flowchart of Fig. 74.

[0601]

Further, when it is determined in step S813 of Fig. 75 that the reference image of the encoding block is a short reference image, the flow proceeds to step S817. In step S817, the vector predicting unit 433 determines whether the reference image of the neighboring block is a long reference image based on the determination result of step S812.

[0602]

When it is determined that the reference image of the neighboring block is a long reference image, the flow returns to step S816. That is, in this case, since the types of the motion (parallax) vectors of the encoding block and the neighboring block are not identical, the motion (parallax) vector of the neighboring block is set to be excluded from the candidate.

[0603]

Moreover, when it is determined in step S817 that the reference image of the neighboring block is a short reference image, the flow proceeds to step S818. In this case, the motion (parallax) vectors of the encoding block and the neighboring block are short reference images. Therefore, in step S818, the vector predicting unit 433 includes the motion (parallax) vector of the neighboring block into the candidate and sets the scaling to be necessary. When the process of step S818 ends, the flow returns to the flowchart of Fig. 74.

[0604]

Next, an example of the flow of a process of generating a candidate motion (parallax) vector from the upper block, executed in step S772 of Fig. 72 will be described with reference to the flowcharts of Figs. 76 and 77.

[0605]

When the process starts, the vector predicting unit 433 selects a block at the position B0 in step S821.

[0606]

In step S822, the vector predicting unit 433 determines whether the block at the position B0 is an

intra mode or not available.

[0607]

When it is determined that the block at the position B0 is neither an intra mode or not available, the flow proceeds to step S823. In step S823, the vector predicting unit 433 determines whether the block at the position B0 refers to the same reference image as the encoding block.

[0608]

When it is determined that the block at the position B0 refers to the same reference image as the encoding block, the flow proceeds to step S824. In step S824, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B0 as the candidate. When the process of step S824 ends, the flow returns to the flowchart of Fig. 72.

[0609]

Moreover, when it is determined in step S822 of Fig. 76 that the block at the position B0 is an intra mode or not available, the flow proceeds to step S825. Moreover, when it is determined in step S823 that the block at the position B0 refers to a reference image different from that of the encoding block, the flow proceeds to step S825.

[0610]

In step S825, the vector predicting unit 433 selects a block at the position B1.

[0611]

In step S826, the vector predicting unit 433 determines whether the block at the position B1 is an intra mode or not available.

[0612]

When it is determined that the block at the position B1 is neither an intra mode or not available, the flow proceeds to step S827. In step S827, the vector predicting unit 433 determines whether the block at the position B1 refers to the same reference image as the encoding block.

[0613]

When it is determined that the block at the position B1 refers to the same reference image as the encoding block, the flow proceeds to step S828. In step S828, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B1 as the candidate. When the process of step S828 ends, the flow returns to the flowchart of Fig. 72.

[0614]

Moreover, when it is determined in step S826 of Fig. 76 that the block at the position B1 is an intra mode or not available, the flow proceeds to step S829. Moreover, when it is determined in step S827 that the block at the position B1 refers to a reference image different from that of the encoding block, the flow proceeds to step S829.

[0615]

In step S829, the vector predicting unit 433 selects a block at the position B2.

[0616]

In step S830, the vector predicting unit 433 determines whether the block at the position B2 is an intra mode or not available.

[0617]

When it is determined that the block at the position B2 is neither an intra mode or not available, the flow proceeds to step S831. In step S831, the vector predicting unit 433 determines whether the block at the position B2 refers to the same reference image as the encoding block.

[0618]

When it is determined that the block at the position B2 refers to the same reference image as the encoding block, the flow proceeds to step S832. In step S832, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B2 as the candidate. When the process of step S832 ends, the flow returns to the flowchart of Fig. 72.

[0619]

Moreover, when it is determined in step S830 of Fig. 76 that the block at the position B2 is an intra mode or not available, the flow proceeds to step S841 of Fig. 77. Moreover, when it is determined in step S831 of Fig. 76 that the block at the position B2 refers to a reference image different from that of the encoding block, the flow proceeds to step S841 of Fig. 77.

[0620]

In step S841 of Fig. 77, the vector predicting unit 433 determines whether a candidate motion (parallax) vector is generated from the left block.

[0621]

In step S842, the vector predicting unit 433 selects a block at the position B0.

[0622]

In step S843, the vector predicting unit 433

determines whether the block at the position B0 is an intra mode or not available.

[0623]

When it is determined that the block at the position B0 is neither an intra mode or not available, the flow proceeds to step S844. In step S844, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B0 as the candidate. When the process of step S844 ends, the flow proceeds to step S851.

[0624]

Moreover, when it is determined in step S843 that the block at the position B0 is an intra mode or not available, the flow proceeds to step S845.

[0625]

In step S845, the vector predicting unit 433 selects a block at the position B1.

[0626]

In step S846, the vector predicting unit 433 determines whether the block at the position B1 is an intra mode or not available.

[0627]

When it is determined that the block at the position B1 is neither an intra mode or not available, the flow proceeds to step S847. In step S847, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B1 as the candidate. When the process of step S847 ends, the flow proceeds to step S851.

[0628]

Moreover, when it is determined in step S846 that the block at the position B1 is an intra mode or not available, the flow proceeds to step S848.

[0629]

In step S848, the vector predicting unit 433 selects a block at the position B2.

[0630]

In step S849, the vector predicting unit 433 determines whether the block at the position B2 is an intra mode or not available.

[0631]

When it is determined in step S849 that the block at the position B2 is an intra mode or not available, the flow returns to the flowchart of Fig. 72.

[0632]

Moreover, when it is determined that the block at the position B2 is neither an intra mode or not available, the flow proceeds to step S850. In step S850, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B2 as the candidate. When the process of step S850 ends, the flow proceeds to step S851.

[0633]

In step S851, the vector predicting unit 433 determines the presence of a scaling process for the motion (parallax) vector of the neighboring block and the presence of the candidate. Since this process is performed in the same manner as that described with reference to the flowchart of Fig. 75, the description thereof will not be provided. In this case, when the process of the flowchart of Fig. 75 ends, the flow returns to the flowchart of Fig. 77 rather than the flowchart of Fig. 74.

[0634]

In step S852, the vector predicting unit 433

determines whether the motion (parallax) vector of the neighboring block is to be excluded from the candidate based on the determination result of step S851.

[0635]

When it is determined that the motion (parallax) vector is to be excluded from the candidate, the flow proceeds to step S853. In step S853, the vector predicting unit 433 excludes the motion (parallax) vector of the upper block from the candidate. When the process of step S853 ends, the flow returns to the flowchart of Fig. 72.

[0636]

Moreover, when it is determined in step S852 of Fig. 77 that the motion (parallax) vector is not to be excluded from the candidate (to be included in the candidate), the flow proceeds to step S854.

[0637]

In step S854, the vector predicting unit 433 determines whether scaling is necessary for the motion (parallax) vector of the neighboring block based on the determination result of step S851.

[0638]

When it is determined that scaling is necessary, the flow proceeds to step S855. In step S855, the vector predicting unit 433 performs a scaling process for the motion (parallax) vector of the neighboring block. When the process of step S855 ends, the flow returns to the flowchart of Fig. 72.

[0639]

Moreover, when it is determined in step S854 of Fig. 77 that scaling is not necessary, the flow returns to the

flowchart of Fig. 72.

[0640]

By executing the respective processes in the above-described manner, the image encoding device 400 can suppress a decrease in the encoding efficiency.

[0641]

[Flow of process during decoding]

In order to correctly decode the encoded data obtained by encoding in the above-described manner, prediction may be performed on the decoding side as the same method as the encoding side. That is, on the decoding side, when the types (short reference picture or long reference picture) of the reference pictures of the encoding vector and the predictive vector are different, the predictive vector is set to be not available.

[0642]

That is, for example, when the type of a reference image of the encoding block is identical to the type of a reference image of a temporal correlation block (that is, when both reference images are short reference images or long reference images), a motion (parallax) vector of the temporal correlation block is used as a candidate. When the types of both reference images are not identical, the vectors are excluded from the candidate. Further, when both the reference image of the encoding block and the reference image of the temporal correlation block are short reference images, scaling of the motion (parallax) vector of the temporal correlation block is performed. When both reference images are long reference images, scaling of the motion (parallax) vector of the temporal correlation block is not performed.

[0643]

Moreover, for example, when the type of the reference image of the encoding block is identical to the type of the reference image of the neighboring block (that is, when both reference images are short reference images or long reference images), the motion (parallax) vector of the neighboring block is used as a candidate. When the types of both reference images are not identical, the vectors are excluded from the candidate. Further, when both the reference image of the encoding block and the reference image of the neighboring block are short reference images, scaling of the motion (parallax) vector of the neighboring block is performed. When both reference images are long reference images, scaling of the motion (parallax) vector of the neighboring block is not performed.

[0644]

By doing so, it is possible to suppress a decrease in the encoding efficiency.

[0645]

An example of the flow of the process during encoding for realizing such control will be described below. Such control can be realized by the image decoding device 500 (Fig. 47) described in the fourth embodiment.

[0646]

The decoding process of the image decoding device 500 is performed in the same manner as that (the second embodiment) described with reference to the flowchart of Fig. 33. Moreover, the prediction process of step S305 of Fig. 33 is performed in the same manner as that

described with reference to the flowchart of Fig. 34.

[0647]

An example of the flow of a PU motion (parallax) vector and reference index generation process executed by the motion compensation unit 512 as the process corresponding to the motion parallax compensation process executed in step S334 of Fig. 34 will be described with reference to the flowchart of Fig. 78.

[0648]

This process is performed basically in the same manner as that of the process (Fig. 67) on the encoding side. However, since the prediction mode is determined on the encoding side, a device on the decoding side may perform the process of the used prediction mode only.

[0649]

When the process starts, in step S861, the lossless decoding unit 302 decodes a prediction mode supplied from the encoding side.

[0650]

In step S862, the mode determining unit 531 determines whether the prediction mode is a merge (skip) mode.

[0651]

When it is determined that the prediction mode is a merge (skip) mode, the flow proceeds to step S863. In step S863, the reference index determining unit 532 and the vector decoding unit 534 performs the process of the merge (skip) mode to generate a motion (parallax) vector and a reference index. When the process of step S863 ends, the flow returns to the flowchart of Fig. 34.

[0652]

Moreover, when it is determined in step S862 of Fig. 78 that the prediction mode is not a merge (skip) mode, the flow proceeds to step S863. In step S864, the vector decoding unit 533 acquires a residual motion (parallax) vector and a reference index. In step S865, the vector decoding unit 533 performs the process of the AMVP mode to generate a predictive motion (parallax) vector. In step S866, the vector decoding unit 533 adds the residual motion (parallax) vector and the predictive motion (parallax) vector. When the process of step S866 ends, the flow returns to the flowchart of Fig. 34.

[0653]

Next, an example of the flow of the merge (skip) mode process executed in step S863 of Fig. 78 will be described with reference to the flowchart of Fig. 79. This process is performed basically in the same manner as the process (Fig. 68) on the encoding side.

[0654]

The respective processes of steps S871 to S874 are executed in the same manner as the respective processes of steps S711 to S714 of Fig. 68. However, the processes of steps S871 and S872 are performed by the reference index determining unit 532, and the processes of steps S873 and S874 are performed by the vector decoding unit 534.

[0655]

In step S875, the vector decoding unit 534 decodes the largest value X of the candidate list.

[0656]

The processes of steps S876 to S880 are executed in the same manner as the processes of steps S716 to S720 of

Fig. 68. However, these processes are executed by the vector decoding unit 534.

[0657]

In step S881, the vector decoding unit 534 decodes the element index of the candidate list. In step S882, the vector decoding unit 534 acquires the motion (parallax) vector and the reference index indicated by the element index. When the process of step S882 ends, the flow returns to the flowchart of Fig. 78.

[0658]

Next, an example of the flow of a process of generating a candidate motion (parallax) vector from the temporal correlation block executed in step S873 of Fig. 79 will be described with reference to the flowchart of Fig. 80. This process is performed basically in the same manner as the process (Fig. 69) on the encoding side.

[0659]

When the process starts, the vector decoding unit 534 decodes the index indicating the temporal correlation picture in step S891.

[0660]

The processes of steps S892 to S902 are performed in the same manner as the processes of steps S732 to S742 of Fig. 69. However, these processes are executed by the vector decoding unit 534.

[0661]

The process of determining the presence of a scaling process for the motion (parallax) vector of the temporal correlation block and the presence of the candidate executed in step S899 of Fig. 80 is executed in the same manner as that described with reference to the

flowchart of Fig. 70, and the description thereof will not be provided. However, in this case, the process is executed by the vector decoding unit 534, and when the process of the flowchart of Fig. 70 ends, the flow returns to the flowchart of Fig. 80.

[0662]

Next, an example of the flow of the AMVP mode process executed in step S865 of Fig. 78 will be described with reference to the flowchart of Fig. 81. This process is performed basically in the same manner as the process (Fig. 71) on the decoding side. The processes of steps S911 to S917 are executed in the same manner as the processes of steps S761 to S767 of Fig. 71. However, these processes are executed by the vector decoding unit 533.

[0663]

In step S918, the vector decoding unit 533 decodes the element index (flag) of the candidate list. In step S919, the vector decoding unit 533 acquires the motion (parallax) vector indicated by the element index. When the process of step S919 ends, the flow returns to the flowchart of Fig. 78.

[0664]

The process of generating a candidate motion (parallax) vector from a spatially neighboring block executed in step S911 of Fig. 81 is executed in the same manner as that described with reference to the flowcharts of Figs. 72 to 77, and the description thereof will not be provided. However, in this case, the process is executed by the vector decoding unit 533, and when the process of the flowchart of Fig. 72 ends, the flow

returns to the flowchart of Fig. 81.

[0665]

Moreover, the process of generating a candidate motion (parallax) vector from the temporal correlation block executed in step S914 of Fig. 81 is executed in the same manner as that described with reference to the flowchart of Fig. 80, and the description thereof will not be provided. However, in this case, the process is executed by the vector decoding unit 533, and when the process of the flowchart of Fig. 80 ends, the flow returns to the flowchart of Fig. 81.

[0666]

By executing the processes in the above-described manner, the image decoding device 500 can correctly decode the encoded data and suppress a decrease in the encoding efficiency.

[0667]

[Summary of handling of motion (parallax) vector]

In the above example, when the motion (parallax) vector is generated from the neighboring block in the AMVP mode, the motion (parallax) vector is retrieved in the order of the positions A0 and A1 for the left neighboring block, for example, and the process ends when the motion (parallax) vector is found. Similarly, the motion (parallax) vector is retrieved in the order of the positions B0, B1, and B2 for the upper neighboring block, for example, and the process ends when the motion (parallax) vector is found. In this case, since whether the motion (parallax) vector can be used as the candidate is determined after that, for example, even when the motion (parallax) vector found at a certain position

cannot be used as the candidate, retrieving of the motion (parallax) vector at the subsequent positions is not performed.

[0668]

A method of generating the motion (parallax) vector from the neighboring block is not limited to this. For example, whether the motion (parallax) vector at all positions A0 and A1 or positions B0, B1, and B2 cannot be used as the candidate may be determined. That is, the presence of the candidate is determined for the neighboring blocks at respective positions and the presence of the scaling process is determined collectively at the end.

[0669]

By doing so, it is possible to determine whether the motion (parallax) vectors at the respective positions are to be included in the candidate although the process becomes more complex than the above-described example. That is, it is possible to retrieve more appropriate candidates and to further improve the encoding efficiency.

[0670]

Fig. 82 is a diagram for describing an example of handling of a neighboring block. In vector prediction, first, whether a motion (parallax) vector of a neighboring block at each position is to be included in a candidate vector is determined as in the table illustrated in A of Fig. 82.

[0671]

That is, for example, when the types of the reference image of the encoding block and the reference image of the neighboring block are identical (that is,

when both reference images are short reference images or long reference images), the motion (parallax) vector of the temporal correlation block is used as a candidate. When the types of both reference images are not identical, the vectors are excluded from the candidate.

[0672]

After the candidate is selected, whether scaling is to be performed on the candidate vector is determined as in the table illustrated in B of Fig. 82.

[0673]

That is, for example, when both the reference image of the encoding block and the reference image of the neighboring block are short reference images, scaling of the motion (parallax) vector of the neighboring block is performed. In other cases, scaling of the motion (parallax) vector of the neighboring block is not performed.

[0674]

[Flow of process during encoding]

An example of the flow of a process of generating a candidate motion (parallax) vector from the left block of this case will be described with reference to the flowcharts of Figs. 83 and 84. Since the processes described with reference to the flowcharts of Figs. 67 to 72 are executed in the same manner as the processes of this case, the description of these processes will not be provided.

[0675]

The processes of steps S921 to S928 of Fig. 83 are executed in the same manner as the processes of steps S781 to S788 of Fig. 73. When the process of step S924

or S928 ends, the process of generating the candidate motion (parallax) vector from the left block ends, and the flow returns to the flowchart of Fig. 72. Moreover, when it is determined in step S926 that the block at the position A1 is an intra mode or not available, or when it is determined in step S927 that the block at the position A1 refers to a reference image different from that of the encoding block, the flow proceeds to step S931 of Fig. 84.

[0676]

In step S931 of Fig. 84, the vector predicting unit 433 selects a block at the position A0.

[0677]

In step S932, the vector predicting unit 433 determines whether the block at the position A0 is an intra mode or not available.

[0678]

When it is determined that the block at the position A0 is neither an intra mode nor unavailable, the flow proceeds to step S933. In step S933, the vector predicting unit 433 determines the presence of the candidate motion (parallax) vector of the block at the position A0.

[0679]

In step S934, the vector predicting unit 433 determines whether the candidate is to be included based on the determination result of step S933.

[0680]

When it is determined that the candidate is to be included, the flow proceeds to step S935. In step S935, the vector predicting unit 433 uses the motion (parallax) vector at the position A0 as the candidate. When the

process of step S935 ends, the flow proceeds to step S943.

[0681]

Moreover, when it is determined in step S934 that the candidate is not to be included, the flow proceeds to step S936. In step S936, the vector predicting unit 433 excludes the motion (parallax) vector of the block at the position A0 from the candidate.

[0682]

When the process of step S936 ends, the flow proceeds to step S937. Moreover, when it is determined in step S932 that the block at the position A0 is neither intra mode nor unavailable, the flow proceeds to step S937.

[0683]

In step S937, the vector predicting unit 433 selects a block at the position A1.

[0684]

In step S938, the vector predicting unit 433 determines whether the block at the position A1 is intra mode or not available. When it is determined that the block at the position A1 is an intra mode or not available, the process of generating the candidate motion (parallax) vector from the left block ends, the flow returns to the flowchart of Fig. 72.

[0685]

Moreover, when it is determined in step S938 of Fig. 84 that the block at the position A1 is neither intra mode nor unavailable, the flow proceeds to step S939. In step S939, the vector predicting unit 433 determines the presence of the candidate motion (parallax) vector of the block at the position A1.

[0686]

In step S940, the vector predicting unit 433 determines whether the candidate is to be included based on the determination result of step S939.

[0687]

When it is determined that the candidate is to be included, the flow proceeds to step S941. In step S941, the vector predicting unit 433 uses the motion (parallax) vector at the position A1 as the candidate. When the process of step S941 ends, the flow proceeds to step S943.

[0688]

Moreover, when it is determined in step S940 that the candidate is not to be included, the flow proceeds to step S942. In step S942, the vector predicting unit 433 excludes the motion (parallax) vector of the block at the position A1 from the candidate. When the process of step S942 ends, the process of generating the candidate motion (parallax) vector from the left block ends, the flow returns to the flowchart of Fig. 72.

[0689]

In step S943 of Fig. 84, the vector predicting unit 433 determines the presence of the scaling process for the motion (parallax) vector of the neighboring block.

[0690]

In step S944, the vector predicting unit 433 determines whether scaling is necessary for the motion (parallax) vector of the neighboring block based on the determination result of step S943.

[0691]

When it is determined that scaling is necessary, the flow proceeds to step S945. In step S945, the vector

predicting unit 433 performs a scaling process on the motion (parallax) vector of the neighboring block. When the process of step S945 ends, the flow returns to the flowchart of Fig. 72.

[0692]

Moreover, when it is determined in step S944 of Fig. 84 that scaling is not necessary, the flow returns to the flowchart of Fig. 72.

[0693]

Next, an example of the flow of a process of determining the presence of a candidate motion (parallax) vector of the neighboring block executed in steps S933 and S939 of Fig. 84 will be described with reference to the flowchart of Fig. 85.

[0694]

The processes of steps S951 to S954 and S957 of Fig. 85 are executed in the same manner as the processes of steps S811 to S814 and S817 of Fig. 75.

[0695]

When it is determined in step S953 that the reference image of the encoding block is a long reference image and it is determined in step S954 that the reference image of the neighboring block is a long reference image, the vector predicting unit 433 sets the motion (parallax) vector of the neighboring block (the block at the position A0 or A1) is to be included in the candidate in step S955.

[0696]

Moreover, when it is determined in step S953 that the reference image of the encoding block is a long reference image and it is determined in step S954 that

the reference image of the neighboring block is a short reference image, or when it is determined in step S953 that the reference image of the encoding block is a short reference image and it is determined in step S957 that the reference image of the neighboring block is a long reference image, since the types of the reference images of the encoding block and the neighboring block are different, the vector predicting unit 433 sets the motion (parallax) vector of the neighboring block (the block at the position A0 or A1) to be excluded from the candidate in step S956.

[0697]

Further, when it is determined in step S953 that the reference image of the encoding block is a short reference image and it is determined in step S957 that the reference image of the neighboring block is a short reference image, the vector predicting unit 433 sets the motion (parallax) vector of the neighboring block (the block at the position A0 or A1) to be included in the candidate in step S958.

[0698]

When the process of step S955, S956, or S958 ends, the flow returns to the flowchart of Fig. 84.

[0699]

Next, an example of the flow of a process of determining the presence of a scaling process for the motion (parallax) vector of the neighboring block executed in step S943 of Fig. 84 will be described with reference to the flowchart of Fig. 86.

[0700]

The processes of steps S961 to S963 and S965 of Fig.

86 are performed in the same manner as the processes of steps S811 to S813 and S817 of Fig. 75.

[0701]

When it is determined in step S963 that the reference image of the encoding block is a long reference image, or when it is determined in step S963 that the reference image of the encoding block is a short reference image and it is determined in step S965 that the reference image of the neighboring block is a long reference image, the flow proceeds to step S964. That is, when it is determined that at least one of the reference image of the encoding block and the reference image of the neighboring block is a long reference image, the flow proceeds to step S964.

[0702]

In step S964, the vector predicting unit 433 sets the scaling to be not necessary. When the process of step S964 ends, the flow returns to the flowchart of Fig. 84.

[0703]

Moreover, when it is determined in step S963 of Fig. 86 that the reference image of the encoding block is a short reference image and it is determined in step S965 that the reference image of the neighboring block is a short reference image (that is, when it is determined that both reference images of the encoding block and the neighboring block are short reference images), the flow proceeds to step S966.

[0704]

In step S966, the vector predicting unit 433 sets the scaling to be necessary. When the process of step

S966 ends, the flow returns to the flowchart of Fig. 84.

[0705]

Next, an example of the flow of a process of generating a candidate motion (parallax) vector from the upper block will be described with reference to the flowcharts of Figs. 87 to 89.

[0706]

The processes of steps S971 to S982 of Fig. 87 are executed in the same manner as the processes of steps S821 to S832 of Fig. 76. When the process of step S974, S978, or S982 ends, the flow proceeds to step S1017 of Fig. 89. Moreover, when it is determined in step S980 that the block at the position B2 is an intra mode or not available, or when it is determined in step S981 that the block at the position B2 refers to a reference image different from that of the encoding block, the flow proceeds to step S991 of Fig. 88.

[0707]

In step S991 of Fig. 88, the vector predicting unit 433 determines whether a candidate motion (parallax) vector is generated from the left block.

[0708]

In step S992, the vector predicting unit 433 selects a block at the position B0.

[0709]

In step S993, the vector predicting unit 433 determines whether the block at the position B0 is an intra mode or not available.

[0710]

When it is determined that the block at the position B0 is neither an intra mode nor unavailable, the

flow proceeds to step S994. In step S994, the vector predicting unit 433 determines the presence of the candidate motion (parallax) vector of the block at the position B0. Since this process is performed in the same manner as that described with reference to the flowchart of Fig. 85, the description thereof will not be provided.

[0711]

In step S995, the vector predicting unit 433 determines whether the candidate is to be excluded based on the determination result of step S994.

[0712]

When the motion (parallax) vector is to be excluded from the candidate, the flow proceeds to step S996. In step S996, the vector predicting unit 433 excludes the motion (parallax) vector of the block at the position B0 from the candidate. When the process of step S996 ends, the flow proceeds to step S998.

[0713]

Moreover, when it is determined in step S995 that the motion (parallax) vector is not to be excluded from the candidate, the flow proceeds to step S997. In step S997, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B0 as the candidate. When the process of step S997 ends, the flow proceeds to step S1017 of Fig. 89.

[0714]

Moreover, when it is determined in step S993 that the block at the position B0 is an intra mode or not available, the flow proceeds to step S998.

[0715]

In step S998, the vector predicting unit 433

selects a block at the position B1.

[0716]

In step S999, the vector predicting unit 433 determines whether the block at the position B1 is an intra mode or not available.

[0717]

When it is determined that the block at the position B1 is neither an intra mode nor unavailable, the flow proceeds to step S1000. In step S1000, the vector predicting unit 433 determines the presence of the candidate motion (parallax) vector of the block at the position B1. Since this process is performed in the same manner as that described with reference to the flowchart of Fig. 85, the description thereof will not be provided.

[0718]

In step S1001, the vector predicting unit 433 determines whether the candidate is to be excluded based on the determination result of step S1000.

[0719]

When the motion (parallax) vector is to be excluded from the candidate, the flow proceeds to step S1002. In step S1002, the vector predicting unit 433 excludes the motion (parallax) vector of the block at the position B1 from the candidate. When the process of step S1002 ends, the flow proceeds to step S1011 of Fig 89.

[0720]

Moreover, when it is determined in step S1001 of Fig. 88 that the motion (parallax) vector is not to be excluded from the candidate, the flow proceeds to step S1003. In step S1003, the vector predicting unit 433 uses the motion (parallax) vector of the block at the

position B1 as the candidate. When the process of step S1003 ends, the flow proceeds to step S1017 of Fig. 89.

[0721]

Moreover, when it is determined in step S999 that the block at the position B1 is an intra mode or not available, the flow proceeds to step S1011 of Fig 89.

[0722]

In step S1011 of Fig. 89, the vector predicting unit 433 selects a block at the position B2.

[0723]

In step S1012, the vector predicting unit 433 determines whether the block at the position B2 is an intra mode or not available.

[0724]

When it is determined that the block at the position B2 is neither an intra mode nor unavailable, the flow proceeds to step S1013. In step S1013, the vector predicting unit 433 determines the presence of the candidate motion (parallax) vector of the block at the position B2. Since this process is performed in the same manner as that described with reference to the flowchart of Fig. 85, the description thereof will not be provided.

[0725]

In step S1014, the vector predicting unit 433 determines whether the candidate is to be excluded based on the determination result of step S1013.

[0726]

When the motion (parallax) vector is to be excluded from the candidate, the flow proceeds to step S1015. In step S1015, the vector predicting unit 433 excludes the motion (parallax) vector of the block at the position B2

from the candidate. When the process of step S1015 ends, the flow proceeds to step S1017.

[0727]

Moreover, when it is determined in step S1014 that the motion (parallax) vector is not to be excluded from the candidate, the flow proceeds to step S1016. In step S1016, the vector predicting unit 433 uses the motion (parallax) vector of the block at the position B2 as the candidate. When the process of step S1016 ends, the flow proceeds to step S1017.

[0728]

In step S1017, the vector predicting unit 433 determines the presence of the scaling process for the motion (parallax) vector of the neighboring block. Since this process is performed in the same manner as that described with reference to the flowchart of Fig. 86, the description thereof will not be provided.

[0729]

In step S1018, the vector predicting unit 433 determines whether scaling is necessary for the motion (parallax) vector of the neighboring block based on the determination result of step S1017.

[0730]

When it is determined that scaling is necessary, the flow proceeds to step S1019. In step S1019, the vector predicting unit 433 performs a scaling process on the motion (parallax) vector of the neighboring block. When the process of step S1019 ends, the flow returns to the flowchart of Fig. 72.

[0731]

Moreover, when it is determined in step S1018 of

Fig. 89 that scaling is not necessary, the flow returns to the flowchart of Fig. 72.

[0732]

By executing the respective processes in the above-described manner, the image encoding device 400 can suppress a decrease in the encoding efficiency.

[0733]

Since the image decoding device 500 performs these processes in the same manner as the image encoding device 400 described above, the description thereof will not be provided. Due to this, the image decoding device 500 can correctly decode the encoded data and suppress a decrease in the encoding efficiency.

[0734]

[Summary of handling of motion (parallax) vector]

In the above, a first example has been described with reference to Figs. 66 to 81, and a second example has been described with reference to Figs. 82 to 89.

[0735]

Besides these examples, for example, in the first example, when both the reference image of the encoding block and the reference image of the neighboring block are long reference images, the motion (parallax) vector of the neighboring block may be used as the candidate only when the encoding block and the neighboring block refer to the same reference images. When the encoding block and the neighboring block refer to different reference images, the motion (parallax) vector of the neighboring block may be excluded from the candidate, and the retrieving process may not be provided.

[0736]

As described above, the long reference image is applied to a fixed region such as a background image and an image of a different view. Thus, even when both the reference image of the encoding block and the reference image of the neighboring block are such long reference images, if the reference images are different images, it is expected that the reference images have low correlation. That is, it is expected that the correlation between the motion (parallax) vectors of the encoding block and the neighboring block are low.

[0737]

Thus, by excluding a pattern (a case where both reference images of the encoding block and the neighboring block are long reference images and both images are different images) in which it is expected that such correlation is low from candidate vectors, it is possible to further suppress a decrease in the encoding efficiency. Moreover, by not providing the process of determining the presence of the candidate and the presence of scaling, it is possible to decrease processing load.

[0738]

Fig. 90 is a diagram for describing another example of handling of a temporal correlation block and a neighboring block. In vector prediction, whether the motion (parallax) vector of the temporal correlation block is to be included into the candidate vector and whether scaling is to be performed are determined as in the table illustrated in A of Fig. 90.

[0739]

That is, this is the same as the case of A of Fig.

66.

[0740]

Moreover, in vector prediction, whether the motion (parallax) vector of the neighboring block is to be included in the candidate vector and whether scaling is to be performed are determined as in the table illustrated in B of Fig. 90.

[0741]

That is, for example, when both reference images of the encoding block and the neighboring block are long reference images, and the motion (parallax) vector of the neighboring block is used as the candidate only when the reference images are the same. Further, when these reference images are different, the motion (parallax) vector of the neighboring block is excluded from the candidate, and the process of determining the presence of the candidate and the presence of scaling is not provided.

[0742]

When both the reference image of the encoding block and the reference image of the neighboring block are short reference images, scaling of the motion (parallax) vector of the neighboring block is performed. When both reference images are long reference images, scaling of the motion (parallax) vector of the neighboring block is not performed.

[0743]

[Flow of process during encoding]

An example of the flow of a process of generating a candidate motion (parallax) vector from the left block of this case will be described with reference to the flowcharts of Figs. 91 and 92. Since the processes

described with reference to the flowcharts of Figs. 67 to 72 are executed in the same manner as the processes of this case, the description of these processes will not be provided.

[0744]

The processes of steps S1031 to S1038 of Fig. 91 are executed basically in the same manner as the processes of steps S781 to S788 of Fig. 73. When the process of step S1034 or S1038 ends, the process of generating the candidate motion (parallax) vector from the left block ends, and the flow returns to the flowchart of Fig. 72. Moreover, when it is determined in step S1035 that the block at the position A1 is an intra mode or not available, the flow proceeds to step S1041 of Fig. 92.

[0745]

However, when it is determined in step S1037 of Fig. 91 that the block at the position A1 refers to a reference image different from that of the encoding block, the flow proceeds to step S1039.

[0746]

In step S1039, the vector predicting unit 433 determines whether both the reference images of the encoding block and the neighboring block are long reference images.

[0747]

When it is determined that both reference images of the encoding block and the neighboring block are long reference images, the process of generating a candidate motion (parallax) vector from the left block ends, the flow returns to the flowchart of Fig. 72.

[0748]

Moreover, when it is determined that at least one of the reference images of the encoding block and the neighboring block is a short reference image, the flow proceeds to step S1041 of Fig. 92.

[0749]

The processes of steps S1041 to S1051 of Fig. 92 are executed in the same manner as the processes of steps S791 to S799 of Fig. 74.

[0750]

That is, when it is determined that both reference images of the encoding block and the neighboring block are long reference images, all processes of the flowchart of Fig. 92 are not provided. Thus, it is possible to decrease processing load.

[0751]

Next, an example of the flow of a process of generating a candidate motion (parallax) vector from the upper block of this case will be described with reference to the flowcharts of Figs. 93 and 94.

[0752]

The processes of steps S1071 to S1082 of Fig. 93 are executed basically in the same manner as the processes of steps S821 to S832 of Fig. 76. When the process of step S1074, S1078, or S1082 ends, the process of generating a candidate motion (parallax) vector from the upper block ends, and the flow returns to the flowchart of Fig. 72. Moreover, when it is determined in step S1080 that the block at the position B2 is an intra mode or not available, the flow proceeds to step S1091 of Fig. 94.

[0753]

However, when it is determined in step S1081 of Fig. 93 that the block at the position B2 refers to a reference image different from that of the encoding block, the flow proceeds to step S1083.

[0754]

In step S1083, the vector predicting unit 433 determines whether both reference images of the encoding block and the neighboring block are long reference images.

[0755]

When it is determined that both reference images of the encoding block and the neighboring block are long reference images, the process of generating a candidate motion (parallax) vector from the upper block ends, the flow returns to the flowchart of Fig. 72.

[0756]

Moreover, when it is determined that at least one of the reference images of the encoding block and the neighboring block is a short reference image, the flow proceeds to step S1091 of Fig. 94.

[0757]

The processes of steps S1091 to S1105 of Fig. 94 are executed in the same manner as the processes of steps S841 to S855 of Fig. 77.

[0758]

That is, when it is determined that both reference images of the encoding block and the neighboring block are long reference images, all processes of the flowchart of Fig. 94 are not provided. Thus, it is possible to decrease processing load.

[0759]

The series of processes can be applied to multi-view image encoding and decoding (multi-view encoder and decoder). That is, the encoding efficiency can be improved when multi-view encoding and decoding are performed. The series of processes can be applied to layer image encoding (space scalability) and layer image decoding (multilayer encoder and decoder). That is, the encoding efficiency can be improved when layer image encoding and decoding are performed.

[0760]

Moreover, the series of processes can be applied to a so-called 2D image of a mono-view image (1-view).

[0761]

The present technique can be applied to an image information encoding device and an image decoding method that are used when image information (a bit stream) which has been compressed by orthogonal transform such as discrete cosine transform and motion compensation as in the case of MPEG, H. 26x, and the like is received via a network medium such as satellite broadcasting, a cable TV, the Internet, or a cellular phone. Moreover, the present technique can be applied to an image encoding device and an image decoding device that are used when the image information (a bit stream) is processed on a storage medium such as an optical or magnetic disk, or a flash memory. Further, the present technique can be applied to a motion prediction compensating device included in the image encoding device, the image decoding device, and the like.

[0762]

[7. Seventh embodiment]

[Computer]

The above-described series of processes can be executed not only by hardware but also by software. When the series of processes are executed by software, a program included in the software is installed in a computer. Here, the computer may be a computer integrated into an exclusive hardware or a general-purpose personal computer which can execute various functions by installing various programs in the computer.

[0763]

In Fig. 95, a CPU (Central Processing Unit) 701 of a personal computer 700 executes various processes according to a program stored in a ROM (Read Only Memory) 702 or a program loaded from a storage unit 713 into a RAM (Random Access Memory) 703. Data necessary for the CPU 701 to execute the various processes is also appropriately stored in the RAM 703.

[0764]

The CPU 701, the ROM 702, and the RAM 703 are connected to one another via a bus 704. An input/output interface 710 is also connected to the bus 704.

[0765]

An input unit 711 formed of a keyboard, a mouse and the like, an output unit 712 formed of a display formed of a CRT (Cathode Ray Tube) and an LCD (Liquid Crystal Display), a speaker, and the like, the storage unit 713 formed of a hard disk and the like, and a communication unit 714 formed of a modem and the like are connected to the input/output interface 710. The communication unit 714 performs a communication process via a network including the Internet.

[0766]

A drive 715 is connected to the input/output interface 710 as needed, a removable medium 721 such as a magnetic disc, an optical disc, a magneto-optical disc, and a semiconductor memory is appropriately mounted thereon, and a computer program read from the medium is installed on the storage unit 713 as necessary.

[0767]

When the series of processes is executed by software, a program included in the software is installed via a network or a recording medium.

[0768]

The recording medium is composed not only of the removable medium 721 including the magnetic disc (including a flexible disc), the optical disc (including a CD-ROM (Compact Disc-Read Only Memory) and a DVD (Digital Versatile Disc)), the magneto-optical disc (including an MD (Mini Disc)), and the semiconductor memory, in which the program is recorded, distributed to a user for distributing the program separately from an apparatus main body but also of the ROM 702 in which the program is recorded and the hard disc included in the storage unit 713 distributed to the user in a state being embedded in advance in the apparatus main body as illustrated in Fig. 95, for example.

[0769]

The program executed by the computer may be a program executing processing in a time-sequential manner in accordance with the procedures described in this specification and may be a program executing the processing in a parallel manner or at necessary times

such as in response to calls.

[0770]

Here, in this specification, the steps that describe the program recorded in the recording medium includes not only processing which is executed in time-sequential manner in accordance with described procedures but also processing which is executed in parallel and/or separately even if it is not always executed in time-sequential manner.

[0771]

In this specification, the term "system" is used to imply an apparatus as a whole, which includes a plurality of devices (apparatuses).

[0772]

In the above description, the configuration described as one apparatus (or processor) may be split into a plurality of apparatuses (or processors). Alternatively, the configuration described as a plurality of apparatuses (or processors) may be integrated into a single apparatus (or processor). Moreover, a configuration other than those discussed above may be included in the above-described configuration of each apparatus (or each processor). If the configuration and the operation of a system as a whole is substantially the same, part of the configuration of an apparatus (or processor) may be added to the configuration of another apparatus (or another processor). The embodiments of the present disclosure are not limited to the above-described embodiments, but various modifications can be made in a range not departing from the gist of the present disclosure.

[0773]

[8. Eighth embodiment]

The image encoding device and the image decoding device according to the above-described embodiments can be applied to various electronic apparatuses such as a transmitter or a receiver that distributes signals on cable broadcasting (such as satellite broadcasting or a cable TV) or on the Internet and distributes signals to a terminal by cellular communication, a recording device that records images on a medium such as an optical disc, a magnetic disk, or a flash memory, or a reproducing device that reproduces images from these storage media. Four application examples will be described below.

[0774]

[First application example: Television apparatus]

Fig. 96 illustrates an example of a schematic configuration of a television apparatus to which the above-described embodiment is applied. A television apparatus 900 includes an antenna 901, a tuner 902, a demultiplexer 903, a decoder 904, a video signal processor 905, a display unit 906, an audio signal processor 907, a speaker 908, an external interface 909, a control unit 910, a user interface 911, and a bus 912.

[0775]

The tuner 902 extracts a signal of a desired channel from a broadcast signal received via the antenna 901 and demodulates the extracted signal. Then, the tuner 902 outputs an encoded bit stream obtained by demodulation to the demultiplexer 903. That is, the tuner 902 serves as transmitting means in the television apparatus 900, which receives the encoded stream in which

the image is encoded.

[0776]

The demultiplexer 903 separates a video stream and an audio stream of a program to be watched from the encoded bit stream and outputs each separated stream to the decoder 904. Moreover, the demultiplexer 903 extracts auxiliary data such as EPG (Electronic Program Guide) from the encoded bit stream and supplies the extracted data to the control unit 910. The demultiplexer 903 may descramble when the encoded bit stream is scrambled.

[0777]

The decoder 904 decodes the video stream and the audio stream input from the demultiplexer 903. Then, the decoder 904 outputs video data generated by a decoding process to the video signal processor 905. Moreover, the decoder 904 outputs audio data generated by the decoding process to the audio signal processor 907.

[0778]

The video signal processor 905 reproduces the video data input from the decoder 904 and allows the display unit 906 to display video. The video signal processor 905 may also allow the display unit 906 to display an application screen supplied via the network. The video signal processor 905 may also perform an additional process such as noise removal, for example, to the video data according to setting. Further, the video signal processor 905 may generate a GUI (Graphical User Interface) image such as a menu, a button, and a cursor, for example, and superimpose the generated image on an output image.

[0779]

The display unit 906 is driven by a drive signal supplied from the video signal processor 905 to display the video or image on a video screen of a display device (for example, a liquid crystal display, a plasma display, an OELD (Organic ElectroLuminescence Display (organic EL display) and the like).

[0780]

The audio signal processor 907 performs a reproducing process such as D/A conversion and amplification on the audio data input from the decoder 904 and allows the speaker 908 to output the audio. The audio signal processor 907 may also perform an additional process such as the noise removal to the audio data.

[0781]

The external interface 909 is the interface for connecting the television apparatus 900 and an external device or the network. For example, the video stream or the audio stream received via the external interface 909 may be decoded by the decoder 904. That is, the external interface 909 also serves as the transmitting means in the television apparatus 900, which receives the encoded stream in which the image is encoded.

[0782]

The control unit 910 includes a processor such as a CPU and a memory such as a RAM and a ROM. The memory stores the program executed by the CPU, program data, the EPG data, data obtained via the network and the like. The program stored in the memory is read by the CPU at startup of the television apparatus 900 to be executed, for example. The CPU controls operation of the

television apparatus 900 according to an operation signal input from the user interface 911, for example, by executing the program.

[0783]

The user interface 911 is connected to the control unit 910. The user interface 911 includes a button and a switch for the user to operate the television apparatus 900, a receiver of a remote control signal and the like, for example. The user interface 911 detects the operation input by the user via the components to generate the operation signal and outputs the generated operation signal to the control unit 910.

[0784]

The bus 912 connects the tuner 902, the demultiplexer 903, the decoder 904, the video signal processor 905, the audio signal processor 907, the external interface 909, and the control unit 910 to one another.

[0785]

In the television apparatus 900 configured in this manner, the decoder 904 has the functions of the image decoding device 50 according to the above-described embodiment. Therefore, when images are decoded in the television apparatus 900, a decrease in the encoding efficiency can be suppressed.

[0786]

[Second application example: Mobile phone]

Fig. 97 illustrates an example of a schematic configuration of a mobile phone to which the above-described embodiment is applied. A mobile phone 920 includes an antenna 921, a communication unit 922, an

audio codec 923, a speaker 924, a microphone 925, a camera unit 926, an image processor 927, a multiplexing/separating unit 928, a recording/reproducing unit 929, a display unit 930, a control unit 931, an operation unit 932, and a bus 933.

[0787]

The antenna 921 is connected to the communication unit 922. The speaker 924 and the microphone 925 are connected to the audio codec 923. The operation unit 932 is connected to the control unit 931. The bus 933 connects the communication unit 922, the audio codec 923, the camera unit 926, the image processor 927, the multiplexing/separating unit 928, the recording/reproducing unit 929, the display unit 930, and the control unit 931 to one another.

[0788]

The mobile phone 920 performs operation such as transmission/reception of an audio signal, transmission/reception of an e-mail or image data, image taking, and recording of data in various operation modes including an audio communication mode, a data communication mode, an imaging mode, and a television-phone mode.

[0789]

In the audio communication mode, an analog audio signal generated by the microphone 925 is supplied to the audio codec 923. The audio codec 923 converts the analog audio signal to the audio data and A/D converts the converted audio data to compress. Then, the audio codec 923 outputs the compressed audio data to the communication unit 922. The communication unit 922

encodes and modulates the audio data to generate a transmission signal. Then, the communication unit 922 transmits the generated transmission signal to a base station (not illustrated) via the antenna 921. Moreover, the communication unit 922 amplifies a wireless signal received via the antenna 921 and applies frequency conversion to the same to obtain a reception signal. Then, the communication unit 922 generates the audio data by demodulating and decoding the reception signal and outputs the generated audio data to the audio codec 923. The audio codec 923 expands the audio data and D/A converts the same to generate the analog audio signal. Then, the audio codec 923 supplies the generated audio signal to the speaker 924 to output the audio.

[0790]

In the data communication mode, for example, the control unit 931 generates character data composing the e-mail according to the operation by the user via the operation unit 932. Moreover, the control unit 931 allows the display unit 930 to display characters. The control unit 931 generates e-mail data according to a transmission instruction from the user via the operation unit 932 to output the generated e-mail data to the communication unit 922. The communication unit 922 encodes and modulates the e-mail data to generate the transmission signal. Then, the communication unit 922 transmits the generated transmission signal to the base station (not illustrated) via the antenna 921. Moreover, the communication unit 922 amplifies the wireless signal received via the antenna 921 and applies the frequency conversion to the same to obtain the reception signal.

Then, the communication unit 922 demodulates and decodes the reception signal to restore the e-mail data and outputs the restored e-mail data to the control unit 931. The control unit 931 allows the display unit 930 to display contents of the e-mail data and allows the storage medium of the recording/reproducing unit 929 to store the e-mail data.

[0791]

The recording/reproducing unit 929 includes an arbitrary readable/writable storage medium. For example, the storage medium may be a built-in storage medium such as the RAM and the flash memory and may be an externally-mounted storage medium such as the hard disc, the magnetic disc, the magneto-optical disc, the optical disc, a USB (Unallocated Space Bitmap) memory, and a memory card.

[0792]

In the imaging mode, for example, the camera unit 926 takes an image of an object to generate the image data and outputs the generated image data to the image processor 927. The image processor 927 encodes the image data input from the camera unit 926 and stores the encoded stream in the storage medium of the recording/reproducing unit 929.

[0793]

Moreover, in the television-phone mode, for example, the multiplexing/separating unit 928 multiplexes the video stream encoded by the image processor 927 and the audio stream input from the audio codec 923 and outputs the multiplexed stream to the communication unit 922. The communication unit 922 encodes and modulates the

stream to generate the transmission signal. Then, the communication unit 922 transmits the generated transmission signal to the base station (not illustrated) via the antenna 921. Moreover, the communication unit 922 amplifies the wireless signal received via the antenna 921 and applies the frequency conversion to the same to obtain the reception signal. The transmission signal and the reception signal may include the encoded bit stream. Then, the communication unit 922 restores the stream by demodulating and decoding the reception signal and outputs the restored stream to the multiplexing/separating unit 928. The multiplexing/separating unit 928 separates the video stream and the audio stream from the input stream and outputs the video stream and the audio stream to the image processor 927 and the audio codec 923, respectively. The image processor 927 decodes the video stream to generate the video data. The video data is supplied to the display unit 930 and a series of images is displayed by the display unit 930. The audio codec 923 expands the audio stream and D/A converts the same to generate the analog audio signal. Then, the audio codec 923 supplies the generated audio signal to the speaker 924 to output the audio.

[0794]

In the mobile phone 920 configured in this manner, the image processor 927 has the functions of the image encoding device 10 and the image decoding device 50 according to the above-described embodiment. Therefore, when images are encoded and decoded in the mobile phone 920, a decrease in the encoding efficiency can be

suppressed.

[0795]

[Third application example: Recording/reproducing apparatus]

Fig. 98 illustrates an example of a schematic configuration of the recording/reproducing apparatus to which the above-described embodiment is applied. The recording/reproducing apparatus 940 encodes the audio data and the video data of a received broadcast program and records the encoded data on the recording medium, for example. Moreover, the recording/reproducing apparatus 940 may encode the audio data and the video data obtained from another apparatus and record the encoded data on the recording medium, for example. Moreover, the recording/reproducing apparatus 940 reproduces the data recorded on the recording medium by a monitor and the speaker according to the instruction of the user. In this case, the recording/reproducing apparatus 940 decodes the audio data and the video data.

[0796]

The recording/reproducing apparatus 940 includes a tuner 941, an external interface 942, an encoder 943, a HDD (Hard Disk Drive) 944, a disc drive 945, a selector 946, a decoder 947, an OSD (On-Screen Display) 948, a control unit 949, and a user interface 950.

[0797]

The tuner 941 extracts a signal of a desired channel from the broadcast signal received via an antenna (not illustrated) and demodulates the extracted signal. Then, the tuner 941 outputs the encoded bit stream obtained by the demodulation to the selector 946. That

is, the tuner 941 serves as the transmitting means in the recording/reproducing apparatus 940.

[0798]

The external interface 942 is the interface for connecting the recording/reproducing apparatus 940 and the external device or the network. The external interface 942 may be an IEEE1394 interface, a network interface, a USB interface, a flash memory interface and the like, for example. For example, the video data and the audio data received via the external interface 942 are input to the encoder 943. That is, the external interface 942 serves as the transmitting means in the recording/reproducing apparatus 940.

[0799]

The encoder 943 encodes the video data and the audio data when the video data and the audio data input from the external interface 942 are not encoded. Then, the encoder 943 outputs the encoded bit stream to the selector 946.

[0800]

The HDD 944 records the encoded bit stream in which content data such as the video and the audio are compressed, various programs and other data on an internal hard disc. The HDD 944 reads the data from the hard disc when reproducing the video and the audio.

[0801]

The disc drive 945 records and reads the data on and from the mounted recording medium. The recording medium mounted on the disc drive 945 may be the DVD disc (DVD-Video, DVD-RAM, DVD-R, DVD-RW, DVD+R, DVD+RW and the like), a Blu-ray (registered trademark) disc and the like,

for example.

[0802]

The selector 946 selects the encoded bit stream input from the tuner 941 or the encoder 943 and outputs the selected encoded bit stream to the HDD 944 or the disc drive 945 when recording the video and the audio. Moreover, the selector 946 outputs the encoded bit stream input from the HDD 944 or the disc drive 945 to the decoder 947 when reproducing the video and the audio.

[0803]

The decoder 947 decodes the encoded bit stream to generate the video data and the audio data. Then, the decoder 947 outputs the generated video data to the OSD 948. Moreover, the decoder 904 outputs the generated audio data to an external speaker.

[0804]

The OSD 948 reproduces the video data input from the decoder 947 to display the video. The OSD 948 may also superimpose the GUI image such as the menu, the button, and the cursor, for example, on the displayed video.

[0805]

The control unit 949 includes the processor such as the CPU and the memory such as the RAM and ROM. The memory stores the program executed by the CPU, the program data and the like. The program stored in the memory is read by the CPU to be executed at startup of the recording/reproducing apparatus 940, for example. The CPU controls operation of the recording/reproducing apparatus 940 according to an operation signal input from the user interface 950, for example, by executing the

program.

[0806]

The user interface 950 is connected to the control unit 949. The user interface 950 includes a button and a switch for the user to operate the recording/reproducing apparatus 940 and a receiver of a remote control signal, for example. The user interface 950 detects operation by the user via the components to generate the operation signal and outputs the generated operation signal to the control unit 949.

[0807]

In the recording/reproducing apparatus 940 configured in this manner, the encoder 943 has the functions of the image encoding device 10 according to the above-described embodiment. Moreover, the decoder 947 has the functions of the image decoding device 50 according to the above-described embodiment. Therefore, when images are encoded and decoded in the recording/reproducing apparatus 940, a decreased in the encoding efficiency can be suppressed.

[0808]

[Fourth application example: Imaging apparatus]

Fig. 99 illustrates an example of a schematic configuration of an imaging apparatus to which the above-described embodiment is applied. An imaging apparatus 960 images an object to generate the image, encodes the image data and records the encoded data on a recording medium.

[0809]

The imaging apparatus 960 includes an optical block 961, an imaging unit 962, a signal processor 963, an

image processor 964, a display unit 965, an external interface 966, a memory 967, a media drive 968, an OSD 969, a control unit 970, a user interface 971, and a bus 972.

[0810]

The optical block 961 is connected to the imaging unit 962. The imaging unit 962 is connected to the signal processor 963. The display unit 965 is connected to the image processor 964. The user interface 971 is connected to the control unit 970. The bus 972 connects the image processor 964, the external interface 966, the memory 967, the media drive 968, the OSD 969, and the control unit 970 to one another.

[0811]

The optical block 961 includes a focus lens, a diaphragm mechanism, and the like. The optical block 961 forms an optical image of the object on an imaging surface of the imaging unit 962. The imaging unit 962 includes an image sensor such as a CCD (charge coupled device) and a CMOS (complementary metal oxide semiconductor) and converts the optical image formed on the imaging surface to an image signal as an electric signal by photoelectric conversion. Then, the imaging unit 962 outputs the image signal to the signal processor 963.

[0812]

The signal processor 963 performs various camera signal processes such as knee correction, gamma correction, and color correction on the image signal input from the imaging unit 962. The signal processor 963 outputs the image data after the camera signal

process to the image processor 964.

[0813]

The image processor 964 encodes the image data input from the signal processor 963 to generate the encoded data. Then, the image processor 964 outputs the generated encoded data to the external interface 966 or the media drive 968. Moreover, the image processor 964 decodes the encoded data input from the external interface 966 or the media drive 968 to generate the image data. Then, the image processor 964 outputs the generated image data to the display unit 965. The image processor 964 may also output the image data input from the signal processor 963 to the display unit 965 to display the image. The image processor 964 may also superimpose data for display obtained from the OSD 969 on the image output to the display unit 965.

[0814]

The OSD 969 generates the GUI image such as the menu, the button, and the cursor, for example, and outputs the generated image to the image processor 964.

[0815]

The external interface 966 is configured as an USB input/output terminal, for example. The external interface 966 connects the imaging apparatus 960 and a printer when printing the image, for example. Moreover, a drive is connected to the external interface 966 as necessary. The removable medium such as the magnetic disc and the optical disc is mounted on the drive, for example, and the program read from the removable medium may be installed on the imaging apparatus 960. Further, the external interface 966 may be configured as a network

interface connected to the network such as a LAN and the Internet. That is, the external interface 966 serves as the transmitting means in the imaging apparatus 960.

[0816]

The recording medium mounted on the media drive 968 may be an arbitrary readable/writable removable medium such as the magnetic disc, the magneto-optical disc, the optical disc, and the semiconductor memory, for example. Moreover, the recording medium may be fixedly mounted on the media drive 968 to form a non-portable storage unit such as a built-in hard disc drive or SSD (Solid State Drive), for example.

[0817]

The control unit 970 includes the processor such as the CPU and the memory such as the RAM and the ROM. The memory stores the program executed by the CPU and the program data. The program stored in the memory is read by the CPU to be executed at startup of the imaging apparatus 960, for example. The CPU controls operation of the imaging apparatus 960 according to the operation signal input from the user interface 971, for example, by executing the program.

[0818]

The user interface 971 is connected to the control unit 970. The user interface 971 includes a button, a switch and the like for the user to operate the imaging apparatus 960, for example. The user interface 971 detects the operation by the user via the components to generate the operation signal and outputs the generated operation signal to the control unit 970.

[0819]

In the imaging apparatus 960 configured in this manner, the image processor 964 has the functions of the image encoding device 10 and the image decoding device 50 according to the above-described embodiment. Therefore, when images are encoded and decoded in the imaging apparatus 960, a decrease in the encoding efficiency can be suppressed.

[0820]

In the present specification, an example in which various types of information such as threshold values are multiplexed into headers and are transmitted from the encoding side to the decoding side has been described. However, a method of transmitting these items of information is not limited to this example. For example, these items of information may be transmitted or recorded as separate data associated with the encoded bit stream rather than being multiplexed into the encoded bit stream. Here, the term "associate" means that the image (or part of the image such as a slice and a block) included in the bit stream and information corresponding to the image can be linked with each other at the time of decoding. That is, the information may be transmitted on a transmission line other than that of the image (or bit stream). Moreover, the information may be recorded on another recording medium (or another recording area of the same recording medium) other than that of the image (or bit stream). Further, the information and the image (or bit stream) may be associated with each other in optional units such as a plurality of frames, one frame, or a part of the frame, for example.

[0821]

While preferred embodiments of the present disclosure have been described in detail with reference to the accompanying drawings, the present disclosure is not limited to the embodiments. Those skilled in the art will readily appreciate that various modifications and changes may be made in the embodiment without departing from the technical spirit as described in the claims. Accordingly, all such modifications and changes are intended to be included within the scope of the present disclosure as defined in the claims.

[0822]

The present technology may include the following constitutions.

(1) An image processing device including:

a predictive vector generating unit that generates a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when encoding the current parallax vector; and

a difference vector generating unit that generates a difference vector between the current parallax vector and the predictive vector generated by the predictive vector generating unit.

(2) The image processing device according to (1), wherein

the predictive vector generating unit generates a predictive vector of the current parallax vector using a parallax vector of a co-located block included in a co-located picture of a time different from a current picture of the same view as a current view.

(3) The image processing device according to (2), wherein

the predictive vector generating unit sets the co-located block to be available when a property of a vector of the current block is identical to a property of a vector of the co-located block.

(4) The image processing device according to (3), wherein

the property of the vector is a type of a vector, and

the predictive vector generating unit sets the co-located block to be available when the property of the vector of the current block is a parallax vector and the property of the vector of the co-located block is a parallax vector.

(5) The image processing device according to (3) or (4), wherein

the predictive motion vector generating unit determines the property of the vector of the current block and the property of the vector of the co-located block using POC (Picture Order Count) that indicates an output order of pictures.

(6) The image processing device according to (5), wherein

the predictive motion vector generating unit determines the property of the vector of the current block and the property of the vector of the co-located block using the POC of the current picture, the POC of a current reference picture referred from the current picture, the POC of the co-located picture, and the POC of a co-located reference picture referred from the co-

located picture.

(7) The image processing device according to (6), wherein

the predictive motion vector generating unit determines that the property of the vector of the current block and the property of the vector of the co-located block are parallax vectors when the POC of the current picture is identical to the POC of the current reference picture referred from the current picture and the POC of the co-located picture is identical to the POC of the co-located reference picture referred from the co-located picture.

(8) The image processing device according to any of (2) to (7), wherein

the predictive vector generating unit sets the co-located block to be not available when the property of the vector of the current block is different from the property of the vector of the co-located block.

(9) The image processing device according to (8), wherein

the property of the vector is a type of a reference picture, and

the predictive vector generating unit sets the co-located block to be not available when the type of the reference picture of the current block is different from the type of the reference picture of the co-located block.

(10) The image processing device according to (8) or (9), wherein

the property of the vector is a type of a reference picture, and

the predictive vector generating unit skips a

process of searching a reference index when the type of the reference picture of the current block is a long reference type and the type of the reference picture of the co-located block is a long reference type.

(11) The image processing device according to any of (1) to (10), wherein

the predictive vector generating unit generates a predictive vector of the current parallax vector using a parallax vector of a reference block included in a picture of the same time as a current picture of a view different from the current view.

(12) The image processing device according to any of (1) to (11), wherein

the predictive vector generating unit scales the reference parallax vector based on a positional relationship between a current picture and a reference picture referred when generating a predictive motion vector to generate a predictive vector of the current parallax vector.

(13) The image processing device according to any of (1) to (12), wherein

the predictive vector generating unit generates a predictive vector of the current motion vector using a reference motion vector referred when generating a predictive motion vector, when encoding the current motion vector of the current block used in prediction using correlation in a temporal direction, and

the difference vector generating unit generates a difference vector between the current motion vector and the predictive vector generated by the predictive vector generating unit.

(14) The image processing device according to (13), wherein

the predictive vector generating unit generates the predictive vector of the current motion vector using a motion vector of the reference block included in a picture of the same time as the current picture of a view different from the current view.

(15) The image processing device according to (13) or (14), wherein

the predictive vector generating unit generates a predictive vector of the current motion vector using a motion vector of a reference block included in a picture of a time different from the current picture of the same view as the current view.

(16) The image processing device according to (15), wherein

the predictive vector generating unit scales the reference motion vector based on a positional relationship between the current picture and a reference picture referred when generating a predictive motion vector to generate a predictive vector of the current motion vector.

(17) The image processing device according to any of (1) to (16), wherein

the predictive vector generating unit generates the predictive vector using a vector of a block located at the same position as the current block in a state where a position of a pixel of a picture of the same time as the current picture of a view different from the current view is shifted.

(18) The image processing device according to (17),

wherein

the predictive vector generating unit sets a shift amount of the image according to a parallax vector of a neighboring region of the current block.

(19) The image processing device according to (18), wherein

the predictive vector generating unit uses a parallax vector in an X-direction, of the neighboring block in which the value of a parallax vector in a Y-direction is not zero as the shift amount.

(20) The image processing device according to (18) or (19), wherein

the predictive vector generating unit uses a value calculated from parallax vectors in an X-direction, of a plurality of the neighboring blocks in which the value of a parallax vector in a Y-direction is not zero as the shift amount.

(21) The image processing device according to (20), wherein

the predictive vector generating unit uses an average value or a median value of the parallax vectors in the X-direction, of the plurality of the neighboring blocks in which the value of the parallax vector in the Y-direction is not zero as the shift amount of the image.

(22) The image processing device according to any of (17) to (21), wherein

the predictive vector generating unit sets the shift amount of the image according to a global parallax vector.

(23) An image processing method of an image processing device, for allowing the image processing device to

execute:

generating a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when encoding the current parallax vector; and

generating a difference vector between the current parallax vector and the generated predictive vector.

(24) An image processing device including:

a predictive vector generating unit that generates a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when decoding the current parallax vector; and

an arithmetic unit that performs an operation of adding the predictive vector generated by the predictive vector generating unit to a difference vector between the current parallax vector and the predictive vector to reconstruct the current parallax vector.

(25) An image processing method of an image processing device, for allowing the image processing device to execute:

generating a predictive vector of a current parallax vector of a current block used in prediction using correlation in a parallax direction using a reference parallax vector referred when generating a predictive motion vector, when decoding the current parallax vector; and

performing an operation of adding the generated

predictive vector to a difference vector between the current parallax vector and the predictive vector to reconstruct the current parallax vector.

(26) An image processing device including:

a predictive vector generating unit that sets a co-located block to be not available when a type of a reference picture of a current block is different from a type of the reference picture of a co-located block included in a co-located picture of a different time from a current picture when encoding a current motion vector of the current block used in prediction using correlation in a temporal direction and generates a predictive vector of the current motion vector using a reference motion vector referred when generating a predictive motion vector; and

a difference vector generating unit that generates a difference vector between the current motion vector and the predictive vector generated by the predictive vector generating unit.

(27) An image processing method of an image processing device, for allowing the image processing device to execute:

setting a co-located block to be not available when a type of a reference picture of a current block is different from a type of the reference picture of a co-located block included in a co-located picture of a different time from a current picture when encoding a current motion vector of the current block used in prediction using correlation in a temporal direction, and generating a predictive vector of the current motion vector using a reference motion vector referred when

generating a predictive motion vector; and

generating a difference vector between the current motion vector and the generated predictive vector.

REFERENCE SIGNS LIST

[0823]

100 Image encoding device

115 Motion parallax prediction/compensation unit

121 Multi-view decoded picture buffer

131 Motion parallax vector search unit

132 Predicted image generating unit

133 Encoded information accumulation buffer

134 Selector

135 Spatial correlation predictive vector generating unit

136 Temporal parallax correlation predictive vector generating unit

137 Selector

138 Encoding cost calculating unit

139 Mode determining unit

151 Current region processor

152 Correlation region processor

153 L1-prediction processor

154 L0-prediction processor

155 Scheme-1 processor

156 Scheme-2 processor

157 Scheme-3 processor

158 Scheme-4 processor

159 Predictive vector generating unit

300 Image decoding device

312 Motion parallax compensation unit

321 Decoded multi-view picture buffer

331 Encoded information accumulation buffer

332 Spatial correlation predictive vector generating unit

333 Temporal parallax correlation predictive vector generating unit

334 Selector

335 Arithmetic unit

336 Predicted image generating unit

400 Image encoding device

415 Motion prediction/compensation unit

421 Base view encoder

433, 434 Vector predicting unit

457 Different picture-based predictive vector generating unit

471 Parallax vector determining unit

472 Inter-view reference vector generating unit

473 Intra-view reference vector generating unit

500 Image decoding device

512 Motion compensation unit

521 Base view decoder

533, 534 Vector decoding unit

553 Different picture-based predictive vector generating unit

571 Parallax vector determining unit

572 Inter-view reference vector generating unit

573 Intra-view reference vector generating unit

CLAIMS

1. An image processing device comprising:

circuitry configured to:

set a candidate of a predictive vector by setting a co-located vector as unavailable as the candidate of the predictive vector for a coding vector in a current block of a current picture where a reference picture of the coding vector is of a different type of two reference picture types than a reference picture of the co-located vector, wherein the co-located vector is in a co-located block included in a co-located picture of a time different from the current picture but of a same view as the current picture, and wherein the two reference picture types comprise a long-term reference picture type and a short-term reference picture type;

reconstruct the coding vector using the predictive vector in the current block according to the candidate of the predictive vector and a difference vector between the coding vector and the generated predictive vector; and

decode the current block using the coding vector.

2. The image processing device according to claim 1, wherein

the circuitry is further configured to set the candidate of the predictive vector using POC (Picture Order Count) that indicates an output order of pictures.

3. The image processing device according to claim 1, wherein

the circuitry is further configured to set the co-located vector as unavailable as the candidate of the predictive vector where the coding vector comprises a motion vector having a short-term reference picture and the co-located vector comprises an inter-view vector having a long-term reference picture.

4. The image processing device according to claim 3, wherein

the circuitry is further configured to set the co-located vector as unavailable as the candidate of the predictive vector where the coding vector comprises an inter-view vector having a long-term reference picture and the co-located vector comprises a motion vector having a short-term reference picture.

5. The image processing device according to claim 1, wherein the circuitry comprises a Central Processing Unit (CPU).

6. An image processing method, comprising:

setting a candidate of a predictive vector by setting a co-located vector as unavailable as the candidate of the predictive vector for a coding vector in a current block of a current picture where a reference picture of the coding vector is of a different type of two reference picture types than a reference picture of the co-located vector, wherein the co-located vector is in a co-located block included in a co-located picture of a time different from the current picture but of a same view as the current picture, and wherein the two reference picture types comprise a long-term reference picture type and a short-term reference picture type;

reconstructing the coding vector using the predictive vector in the current block according to the candidate of the predictive vector and a difference vector between the coding vector and the generated predictive vector; and

decoding the current block using the coding vector.

7. The image processing method according to claim 6, wherein the candidate of the predictive vector is set using POC (Picture Order Count) that indicates an output order of pictures.

8. The image processing method according to claim 6, wherein the co-located vector is set as unavailable as the candidate of the predictive vector where the coding vector comprises a motion vector having a short-term reference picture and the co-located vector comprises an inter-view vector having a long-term reference picture.

9. The image processing method according to claim 8, wherein the co-located vector is set as unavailable as the candidate of the predictive vector where the coding vector comprises an inter-view vector having a long-term reference picture and the co-located vector comprises a motion vector having a short-term reference picture.