Clean

of PU partition types. Parts (a) to (h) of Fig. 4 illustrate partition shapes in the PU partition types of 2N×2N, 2N×N, 2N×nU, 2N×nD, N×2N, nL×2N, nR×2N and N×N, respectively.

[Fig. 5] Fig. 5 is a diagram illustrating a specific example configuration of a PU size table in which the numbers of PUs and PU sizes are defined in association with CU sizes and PU partition types.

[Fig. 6] Fig. 6 is a diagram illustrating a 2N×N CU and a 2N×nU CU in which an edge having an inclination is present.

[Fig. 7] Fig. 7 is a table illustrating an example of binarization information that defines associations between combinations of CU prediction types and PU partition types and bin sequences.

[Fig. 8] Fig. 8 is a diagram illustrating an example of the binarization information that defines a CU having an 8×8 size.

[Fig. 9] Fig. 9 is a diagram illustrating another example of the binarization information that defines a CU having an 8×8 size.

[Fig. 10] Fig. 10 is a table illustrating another example of the binarization information that defines associations between combinations of CU prediction types and PU partition types and bin sequences.

[Fig. 11] Fig. 11 is a table illustrating another example of the binarization information that defines prediction restriction on a basic inter PU, bi-prediction restriction on a merge PU, skipped derivation of bi-predictive merge candidates are uniformly applied to PUs having 4×4, 4×8, and 8×4 sizes, parts (b) and (c) are diagrams illustrating an example in which the restriction of bi-prediction is imposed only on a basic inter PU without the application of the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates, and part (d) is a diagram illustrating an example in which the bi-prediction restriction on the basic inter PU is uniformly applied to PUs having 4×4, 4×8, and 8×4 sizes and in which the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates are applied to PUs having an 8×8 size.

[Fig. 53] Fig. 53 includes diagrams depicting an example of bi-prediction restriction methods, in which part (a) is a diagram illustrating an example in which the bi-prediction restriction on the basic inter PU and the skipped derivation of bi-predictive merge candidates are applied to 4×4, 4×8, 8×4, and 8×8, and part (b) is a diagram illustrating an example in which the skipped derivation of bi-predictive merge candidates is applied to 4×4, 4×8, 4×8, and 8×8.

[Fig. 54] Fig. 54 is a block diagram illustrating a configuration of a basic motion compensation parameter included in the target prediction tree.

[0121]

The CU information decoding unit 11 supplies the PT information PTI obtained for the target CU to the PU information decoding unit 12. Further, the CU information decoding unit 11 supplies the TT information TTI obtained for the target CU to the TU information decoding unit 13.

[0122]

[PU information decoding unit]

The PU information decoding unit 12 performs a decoding process on the PT information PTI supplied from the CU information decoding unit 11, using the decoding module 10 on the PU level. Specifically, the PU information decoding unit 12 decodes the PT information PTI using the following procedure.

[0123]

The PU information decoding unit 12 refers to the PU partition type information PartMode, and determines the PU partition type for the target prediction tree. Then, the PU information decoding unit 12 sequentially designates each of the PUs included in the target prediction tree as a target PU, and executes the decoding process on the PU information corresponding to the target PU.

[0124]

That is, the PU information decoding unit 12 decodes is reduced and throughput is increased. In addition, a memory for accumulating probabilities of occurrence (states) corresponding to contexts is not necessary. Coding with a fixed probability of 0.5 may be referred to as EP coding (equal probabilities, equal probability coding) or bypass.

[0171]

The operations and effects of the configuration described above will be described with reference to Fig. 6. A context is effective for the improvement in coding efficiency when the same code appears consecutively in a specific condition. Coding efficiency is improved by decoding the suffix portion by referring to contexts when, specifically, 2N×N, 2N×nU, or 2N×nD is consecutively selected in a state where a landscape-oriented partition is selected. This effect works, for example, when 2N×N is selected in a prediction unit subsequent to the prediction unit in which 2N×N was selected.

[0172]

On the other hand, partitions are generally set so as not to lie over edge boundaries, as illustrated in Fig. 6.

[0173]

Specifically, as illustrated in Fig. 6, if an edge E1 having an inclination is present in a region, the PU partition type of a CU 10 and a CU 20 is determined so that no partitions lie across the edge E1.

In the association table BT1, two CU prediction types, namely, the intra CU described above (labeled as "Intra") and inter CU (labeled as "Inter"), are defined for the definition of the respective CU sizes. PU partition types are further defined for the respective CU prediction types.

[0192]

Details are as follows. First, for the intra CU, two PU partition types, namely, 2N×2N and N×N, are defined.

[0193]

A description of 2N×2N will be given hereinafter. In the non-8×8 CU 1012B, only the prefix portion is defined and the bin sequence is "000". The suffix portion is not coded. In the 8×8 CU 1012A, the prefix portion is "000" and the suffix portion is "0".

[0194]

For N×N, on the other hand, a definition is provided only for the non-8×8 CU 1012B. In this case, the prefix portion is "000", and the suffix portion is "1".

[0195]

In this manner, for the intra CU, the prefix portion is "000", which is common.

[0196]

For the inter CU, seven PU partition types, namely, 2N×2N, 2N×N, 2N×nU, 2N×nD, N×2N, nL×2N, and nR×2N, are defined.

[0341]

Part (b) of Fig. 33 illustrates an example of a syntax table in a case that the inter prediction flag is a ternary flag. If a combined list is used, two types, namely, Pred_LC, which means uni-prediction in which one reference frame in an LC list is used, and Pred_Bi, which means bi-prediction, are identified from each other by inter_pred_flag. Otherwise, three types, namely, Pred_L0, which means uni-prediction with the L0 list, Pred_L1, which means uni-prediction with the L1 list, and Pred_Bi, which means bi-prediction, are identified from one another. If the slice is a B slice and bi-prediction is active (DisableBiPred = false), the encoded data includes a first inter prediction flag inter_pred_flag0 for specifying uni-prediction and bi-prediction. If bi-prediction is not active, only in a case that a combined list is not used, the encoded data includes a second inter prediction flag inter_pred_flag1 for specifying uni-prediction and bi-prediction to specify a reference list. The case that a combined list is not used is determined specifically using !UsePredRefLC && !NoBackPredFlag, as illustrated in part (a) of Fig. 33. That is, the determination is based on a flag UsePredRefLC (indicating that a combined list is used if the value of UsePredRefLC is true) specifying whether or not to use a combined list, and a flag NoBackPredFlag (indicating that backward prediction is not used if the value of NoBackPredFlag is true) specifying whether or not to use backward prediction. If a combined list is used, the use of a combined list is determined without list selection. No use of backward prediction means the disabling of Pred_L1. In this case, it may be determined that the list used also when the second inter prediction flag inter_pred_flag1 is not encoded is the combined list (Pred_LC) or the L0 list (Pred_L1). The expression "NoL1PredFlag", which means no use of the L1 list, may be used instead of NoBackPredFlag.

[0342]

A threshold value used to determine whether or not to impose the restriction of bi-prediction or to determine the PU size in a case that the restriction of bi-prediction is imposed may be included in the encoded data. Fig. 34 illustrates an example of a syntax table for bi-prediction restriction. Part (a) of Fig. 34 illustrates the case that the sequence parameter set includes the flag disable_bipred_in_small_PU restricting whether or not to impose the restriction of bi-prediction. As illustrated in Part (a) of Fig. 34, a flag for the restriction of bi-prediction may be encoded independently from a flag disable_inter_4x4 prohibiting a small size PU (here, a 4×4 size PU). The purpose of the flag prohibiting a small size PU is also to reduce the amount of worst-case processing to

1212J. Although not illustrated in Fig. 43, the neighboring merge candidate derivation unit 1212A and the temporal merge candidate derivation unit 1212B are supplied with decoding parameters for an already decoded CU and PU, which are stored in the frame memory 16, particularly, motion compensation parameters on a per-PU basis. In the following, the neighboring merge candidate derivation unit 1212A, the temporal merge candidate derivation unit 1212B, the unique candidate derivation unit 1212C, the combined bi-predictive merge candidate derivation unit 1212D, the non-scaled bi-predictive merge candidate derivation unit 1212E, and the zero vector merge candidate derivation unit 1212F are collectively referred to as "merge candidate deriving means".

[0359]

In the merge motion compensation parameter derivation unit 1212, the merge candidate derivation control unit 1212G controls each merge candidate deriving means to derive a predetermined number MRG_MAX_NUM_CANDS of merge candidates, and stores the derived merge candidates in the merge candidate storage unit 1212H. Here, each merge candidate is composed of prediction list utilization flags predFlagL0 and predFlagL1, which are motion compensation parameters of a PU, reference index numbers refIdxL0 and refIdxL1, and motion vectors mvL0 and mvL1. The merge candidate storage unit 1212H stores sets of the motion compensation parameters described above reference picture (S102). In S103, a redundant merge candidate is removed from among the derived merge candidates A0 to T, and the remaining merge candidates are stored in the merge candidate storage unit 1212H. If the number of non-redundant merge candidates is greater than or equal to MRG_MAX_NUM_CANDS, the derivation of merge candidates is terminated (YES in S104). Otherwise (NO in S104), the operation proceeds to S105. In the case of a B slice (YES in S105), the operation proceeds to S106, and otherwise (NO in S105), the operation skips S106, S107 and S108 and proceeds to S109 (S105). Also in a case that the restriction of bi-prediction is imposed, here, in the case of a small size PU, which corresponds to the case that the bi-directional derivation of merge candidates is skipped, the operation skips the bi-predictive motion candidate derivation process of S107 and S108, and proceeds to S109 (S106). In S107, the combined bi-predictive merge candidate derivation unit 1212D derives combined bi-predictive merge candidates, and the derived combined bi-predictive merge candidates are stored in the merge candidate storage unit 1212H. In S108, the non-scaled bi-predictive merge candidate derivation unit 1212E derives non-scaled bi-predictive merge candidates, and the derived non-scaled bi-predictive merge candidates are stored in the merge candidate storage unit 1212H. Here, if the number of merge candidates is greater than or equal to plurality of merge candidates. This similarly applied to the other merge candidate derivation units described hereinafter.

[0364]

In a case that a neighboring block is not available (unavailable) or in the case of an intra block, the corresponding merge candidates are not derived. A neighboring block is not available in a case that the block is located outside the screen, the block is located outside the slice, or the block is an undecoded block in the scan order of blocks. The positions A0 to B2 may be expressed as follows, where the upper left coordinates of a PU is represented by (xP, yP) and the PU has sizes nPSW and nPSH.

A0: (xP - 1, yP + nPSH)

A1: (xP - 1, yP + nPSH - 1)

B0: (xP + nPSH, yP - 1)

B1: (xP + nPSH - 1, yP - 1)

B2: (xP - 1, yP - 1)

If all the merge candidates corresponding to the positions A0, A1, B0, and B1 are successfully derived, the merge candidate corresponding to the position B2 is not derived. In the derivation of merge candidates for the PU partition type of 2N×N or N×2N and the PU index of 1, the following operation is performed. Only in a case that the motion compensation parameters for each merge candidate do refIdxL0 that occupies substantially the same spatial position as the spatial position of the target PU in the current picture or by copying motion compensation parameters of a reference picture PU specified by the reference index number refIdxL0. The method for deriving the reference index number refIdxL0 and the reference index number refIdxL1 will be described with reference to part (b) of Fig. 46. The reference index number refIdxLX (where X is 0, 1, or C) is determined using the reference pictures refIdxLXA, refIdxLXB, and refIdxLXC of neighboring PUs A, B, and C of the target PU as follows.

(1) In the case of refIdxLXA = refIdxLXB = refIdxLXC,
if refIdxLXA = -1, refIdxLX = 0
otherwise, refIdxLX = refIdxLXA

(2) In the case of refIdxLXA = refIdxLXB,
if refIdxLXA = -1, refIdxLX = refIdxLXC
otherwise, refIdxLX = refIdxLXA

(3) In the case of refIdxLXB = refIdxLXC,
if refIdxLXB = -1, refIdxLX = refIdxLXA
otherwise, refIdxLX = refIdxLXB

(4) In the case of refIdxLXA = refIdxLXC,
if refIdxLXA = -1, refIdxLX = refIdxLXB
otherwise, refIdxLX = refIdxLXA

(5) In the case of refIdxLXA = -1,
refIdxLX = min (refIdxLXB, refIdxLXC)

derivation unit 1213B, a zero vector merge candidate derivation unit 1213F, a motion vector candidate derivation control unit 1213G, a motion vector candidate storage unit 1213H, a motion vector candidate selection unit 1213I, and a motion vector restoration unit 1213J. In the following, the neighboring motion vector candidate derivation unit 1213A, the temporal motion vector candidate derivation unit 1213B, and the zero vector merge candidate derivation unit 1213F are collectively referred to as "motion vector/merge candidate deriving means".

[0376]

In the basic motion compensation parameter derivation unit 1213, the motion vector candidate derivation control unit 1213G controls each motion vector/merge candidate deriving means to derive a predetermined number PMV_MAX_NUM_CANDS of predictive motion vector candidates, and stores the derived predictive motion vector candidates in the motion vector candidate storage unit 1213H. Here, each predictive motion vector candidate is composed of motion vectors mvL0 and mvL1. The motion vector candidate storage unit 1213H stores the combinations of motion compensation parameters described above as predictive motion vector candidates. The stored predictive motion vector candidates are managed as lists (predictive motion vector candidate lists) in which the predictive motion vector candidates are operation without decoding the inter prediction flag inter_pred_flag. If the PU size is a size other than the small PU size (if DisableBiPred != true) (NO in S142), the inter prediction flag decoding unit 1028 decodes the combined inter prediction reference index combined_inter_pred_ref_idx through S143, S144, and S145. If the PU size is the small PU size (YES in S142), the inter prediction flag decoding unit 1028 decodes the combined inter prediction reference index combined_inter_pred_ref_idx through S146, S147, and S148.

[0409]

In S143 and S146, a maximum value MaxPredRef is calculated. The maximum value MaxPredRef is as illustrated in the table TBL37 and the pseudo code CODE37 in part (d) of Fig. 37. Specifically, the maximum value MaxPredRef for a size other than the small size PU, that is, for no restriction of bi-prediction (DisableBiPred != true), is calculated by NumPredRefLC+NumPredRefL0*NumPredRefL1 or NumPredRefL0+NumPredRefL0*NumPredRefL1, that is, by determining the sum of the number of uni-predictive combined reference picture sets (NumPredRefLC or NumPredRefL0) and the number of bi-predictive combined reference picture sets (NumPredRefL0*NumPredRefL1) (S143). The maximum value MaxPredRef for restriction of bi-prediction (DisableBiPred = true) is calculated by NumPredRefLC or NumPredRefL0, that is, encoding process of combined_inter_pred_ref_idx in a case that a variable table is used.

[0412]

In the foregoing, a description has been given of the following methods for reducing the amount of processing for a small PU size: bi-prediction restriction on the basic inter PU (change in the method for decoding the inter prediction flag and the combined inter prediction reference index), bi-prediction restriction on the merge PU (bi-prediction/uni-prediction conversion in the derivation of merge candidates), and skipped calculation of bi-predictive merge candidates. These restrictions may be used individually, or PU sizes to be subject to these restrictions may have different values. Fig. 52 and Fig. 53 illustrate an example of the reduction in the amount of processing for bi-prediction. In Fig. 52 and Fig. 53, open circles indicate that the processes are performed, and crosses indicate that the processes are not performed.

[0413]

Part (a) of Fig. 52 illustrates an example in which the bi-prediction restriction on the basic inter PU, the bi-prediction restriction on the merge PU, and the skipped derivation of bi-predictive merge candidates are uniformly applied to the PUs having the sizes of 4×4, 4×8, and 8×4. Parts (b) and (c) of Fig. 52 illustrate an example in which the restriction of bi-prediction is imposed only on the basic inter PU without the application of the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates. In general, the restriction of bi-prediction for the merge PU may cause a reduction in coding efficiency. It is thus appropriate that the restriction of bi-prediction is imposed only on the basic inter PU.

[0414]

Part (d) of Fig. 52 illustrates an example in which the bi-prediction restriction on the basic inter PU is uniformly applied to the PUs having sizes of 4×4, 4×8, and 8×4 and in which the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates are applied to the PUs having an 8×8 size. Relaxing the bi-prediction restriction on the merge PU compared to the bi-prediction restriction on the basic inter PU will be appropriate in terms of coding efficiency.

[0415]

Part (a) of Fig. 53 illustrates an example in which the bi-prediction restriction on the basic inter PU and the skipped derivation of bi-predictive merge candidates are applied to 4×4, 4×8, 8×4, and 8×8. Simplifying the derivation of merge candidates used as motion compensation parameters of a merge PU, without the restriction of bi-prediction on the merge PU, may reduce the amount of processing regarding bi-prediction in the merge PU. Part (b) of Fig. 53 illustrates an example in which the skipped derivation of bi-predictive merge candidates is applied to 4×4, 4×8, 8×4, and 8×8. In this manner, the skipped derivation of bi-predictive merge candidates may be used alone.

[0416]

In order to implement the cases described above, determination methods may be managed using different flags. For example, flags DisableBiPredFlag, DisableBiPredMerge, and DisableBiPredMergeDerive for imposing the respective prediction restrictions are provided, and are made feasible through the following operation.

[0417]

For example, the bi-prediction restricted PU determination unit 1218 individually derives three flags DisableBiPredFlag, DisableBiPredMerge, and DisableBiPredMergeDerive. In the example illustrated in part (d) of Fig. 52, the flags may be derived as follows.

[0418]

DisableBiPredFlag = (log2CUSize == 3 && PU partition type != 2Nx2N) ? true : false

DisableBiPredMerge, DisableBiPredMergeDerive = ((log2CUSize == 4 && PU partition type == NxN) || log2CUSize

size) is 3, and is not particularly restricted, whereas the minimum PU size is 8×4 or 4×8. That is, the use of 4×4 PU is disabled. In addition, the minimum bi-prediction PU size is 8×4 or 4×8, and the use of bi-prediction for 4×4 PU is disabled. If the level level_idc is greater than or equal to the predetermined threshold value TH2, the logarithmic value of the minimum CU size is 4, and the minimum PU size is restricted to 8×8. That is, the use of 8×4 PU, 4×8 PU, and 4×4 PU is disabled. In addition, the minimum bi-prediction PU size is 16×8, and the use of bi-prediction for 8×8 PU is disabled. The use of bi-prediction for 8×4 PU, 4×8 PU, and 4×4 PU is also disabled due to the restriction of the minimum PU size. It is appropriate that the threshold value TH1 is level 3.1, which is on the line of 720P, and the threshold value TH2 is level 5, which is on the line equivalent to 2560×1600. However, other threshold values may be used.

[0446]

Fig. 85 illustrates another example of level limits of the present invention. The illustrated example is substantially the same as the example in Fig. 84, except that if the level level_idc is less than a predetermined threshold value TH0, the logarithmic value of the minimum CU size, the minimum PU size, and the minimum bi-prediction PU size are 3, 4×4, and 4×4, respectively, and no restrictions restrict_bipred_flag may take one of 0, 1, and otherwise (for example, 2). The following is a description of the individual steps S in the pseudo code illustrated in Fig. 68.

[0506]

S681: If restrict_bipred_flag is equal to "0", the bi-prediction restricted PU determination unit 1218A sets "0" in the variable DisableBipred.

[0507]

S682: If restrict_bipred_flag is equal to "1", the bi-prediction restricted PU determination unit 1218A sets the variable DisableBipred as follows.

[0508]

The restriction of bi-prediction is performed "in a case that the logarithmic value (log2CUSize) of the CU size matches Log2MinCUSize and the PU mode is other than 2N×2N" or "in a case that the logarithmic value (log2CUSize) of the CU size is smaller than Log2MinCUSize".

[0509]

S682: If restrict_bipred_flag is equal to a value other than the values described above (for example, "2"), the bi-prediction restricted PU determination unit 1218A sets the variable DisableBipred as follows.

imposed by the bi-/uni-prediction conversion process is reduced, compared to the case that the bi-/uni-prediction conversion process is performed on all the merge candidates. Furthermore, a configuration in which bi-/uni-prediction conversion is performed on at least one temporal merge candidate ensures that, even if the restriction of bi-prediction is imposed on 8×8 PU having a relatively large PU size, temporal merge candidates for 8×8 PU are uni-predictive, or available. Accordingly, the reduction in coding efficiency may be minimized. In addition, the merge candidate derivation process and the bi-/uni-prediction conversion process are executed in parallel, resulting in processing being efficiently performed.

[0554]

(Modifications)

Preferred modifications 1 to 3 of the present configuration will be described.

[0555]

(Modification 1)

The bi-prediction/uni-prediction conversion unit 1219A may be configured to perform a bi-prediction/uni-prediction conversion process after all the merge candidates have been derived and a merge candidate list has been stored in the merge candidate storage unit 1212H.

[0556]

As already discussed, the configuration illustrated in Fig. 61 and Fig. 71 is a configuration in which the merge candidate derivation process and the bi-/uni-prediction conversion process are executed in parallel.

[0568]

A timetable of the series of processes is as illustrated in, for example, Fig. 74. In a time chart illustrated in Fig. 74, merge candidates A to E have been derived. Among them, two merge candidates are subjected to bi-/uni-prediction conversion by bi-prediction restriction. It is assumed that the time taken to derive the merge candidates C and E is longer than the time taken to derive the merge candidates A, B, and D.

[0569]

As illustrated in Fig. 74, the merge candidates A and B are the first two merge candidates among the merge candidates A to E, and are thus targets of bi-/uni-prediction conversion. The bi-/uni-prediction conversion of the merge candidates A and B is performed in parallel to the execution of the merge candidates C and E, which may require a longer processing time. In the example illustrated in Fig. 74, thus, a list creation process is started upon the completion of the derivation process for the merge candidates C and E, and the entire process ends when the list creation process is completed.

integers will be disclosed with reference to Fig. 76. In the configuration illustrated in Fig. 76, the bi-prediction/uni-prediction conversion unit 1219A in the configuration illustrated in Fig. 61 is replaced by a motion-vector-to-integer conversion unit 1220.

[0590]

The motion-vector-to-integer conversion unit 1220 converts at least one component among one or more non-integer components included in non-integer motion vectors into an integer component. The conversion performed by the motion-vector-to-integer conversion unit 1220 is hereinafter referred to as the conversion of motion vectors into integers.

[0591]

More specifically, in a case that the restriction of bi-prediction is to be imposed, if the merge candidates input from the neighboring merge candidate derivation unit 1212A or the temporal merge candidate derivation unit 1212B include bi-predictive merge candidates, the motion-vector-to-integer conversion unit 1220 determines whether or not the two motion vectors of the bi-predictive merge candidates are non-integer motion vectors. If at least one of the two motion vectors of the bi-predictive merge candidates is a non-integer motion vector, the motion-vector-to-integer conversion unit 1220 converts the non-integer motion vector configured to encode an input image #10 to generate encoded data #1, and to output the encoded data #1.

[0722]

(Configuration of video encoding device)

First, an example configuration of the video encoding device 2 will be described with reference to Fig. 25. Fig. 25 is a functional block diagram illustrating a configuration of the video encoding device 2. As illustrated in Fig. 25, the video encoding device 2 includes an encoding setting unit 21, a dequantization/inverse transform unit 22, a prediction image generation unit 23, an adder 24, a frame memory 25, a subtractor 26, a transform/quantization unit 27, an encoded data generation unit (encoding means) 29, and a PU information generation unit 30.

[0723]

The encoding setting unit 21 generates image data and various kinds of setting information concerning encoding, on the basis of the input image #10.

[0724]

Specifically, the encoding setting unit 21 generates the following image data and setting information.

[0725]

First, the encoding setting unit 21 sequentially splits the input image #10 into slices and tree blocks to generate a CU image #100 for a target CU.

The subtractor 26 subtracts the prediction image Pred from the CU image #100 to generate a prediction residual D for the target CU. The subtractor 26 supplies the generated prediction residual D to the transform/quantization unit 27.

[0736]

The transform/quantization unit 27 applies orthogonal transform and quantization to the prediction residual D to generate a quantized prediction residual. The term "orthogonal transform", as used herein, refers to an orthogonal transform from the pixel domain to the frequency domain. Examples of the orthogonal transform include DCT transform (Discrete Cosine Transform) and DST transform (Discrete Sine Transform).

[0737]

Specifically, the transform/quantization unit 27 refers to the CU image #100 and the CU information CU', and determines the pattern in which the target CU is split into one or a plurality of blocks. The transform/quantization unit 27 splits the prediction residual D into prediction residuals for the respective blocks in accordance with the determined partition pattern.

[0738]

Further, the transform/quantization unit 27 applies orthogonal transform to the prediction residual for each block to generate a prediction residual in the frequency domain. Then, the transform/quantization unit 27 quantizes the prediction residual in the frequency domain to generate a quantized prediction residual for each block.

[0739]

The transform/quantization unit 27 further generates TT setting information TTI' including the generated quantized prediction residual for each block, TT split information for specifying the partition pattern of the target CU, and information concerning all possible patterns of splitting the target CU into individual blocks. The transform/quantization unit 27 supplies the generated TT setting information TTI' to the dequantization/inverse transform unit 22 and the encoded data generation unit 29.

[0740]

The PU information generation unit 30 encodes the PT setting information PTI' if the prediction type indicated by the PT setting information PTI' is inter prediction, and derives PT setting information PTI. The PU information generation unit 30 further generates PTI setting information PTI' on merge candidates, and supplies the PTI setting information PTI' to the encoding setting unit 21.

[0741]

The encoded data generation unit 29 encodes the header information H', the TT setting information TTI', and the PT setting information PTI'. The encoded data generation unit compensation parameter derivation unit 1212", and "the reference frame setting information storage unit 123" in the description of the example configuration [2-3-3] should be substituted with "the motion compensation parameter generation unit 301", "the merge motion compensation parameter generation unit 3012", and "the configuration corresponding to the reference frame setting information storage unit 123", respectively.

[0788]

[2-3-4]' Conversion of motion vectors into integers

The PU information generation unit 30 may convert motion vectors into integers. A specific configuration is as illustrated in Fig. 83. In the configuration illustrated in Fig. 83, the bi-prediction/uni-prediction conversion unit 1219 in the PU information generation unit 30 illustrated in Fig. 55 or 56 is replaced by a bi-prediction/uni-prediction conversion unit 1219A. In the configuration illustrated in Fig. 83, the bi-prediction/uni-prediction conversion unit 1219 in the PU information generation unit 30 illustrated in Fig. 55 or 56 is replaced by a motion-vector-to-integer conversion unit 1220.

[0789]

More details are similar to those described in, for example, the description of the example configuration [2-3-4] of the video decoding device 1, and a description thereof

[0815]

(Processing flow)

The CU encoding process of the video encoding device 2 will be described hereinafter with reference to Fig. 26. In the following, it is assumed that a target CU is an inter CU or a skip CU. Fig. 26 is a flowchart illustrating an example of the flow of the CU encoding process (inter/skip CU) of the video encoding device 2.

[0816]

When the CU encoding process starts, the encoding setting unit 21 determines CU prediction information on the target CU, and the encoded data generation unit 29 encodes the CU prediction information determined by the encoding setting unit 21 (S21). This process is performed on a per-CU basis.

[0817]

Specifically, the encoding setting unit 21 determines whether or not the target CU is a skip CU. If the target CU is a skip CU, the encoding setting unit 21 causes the encoded data generation unit 29 to encode the skip flag SKIP. If the target CU is not a skip CU, the encoding setting unit 21 causes the encoded data generation unit 29 to encode the CU prediction type information Pred_type.

[0818]

Then, processing is performed on a per-PU basis.

Disc: registered trademark).

[0837]

The recording apparatus PROD_C may further include sources from which moving images to be input to the encoder PROD_C1 are supplied, including a camera PROD_C3 for capturing a moving image, an input terminal PROD_C4 through which a moving image is input from outside, a receiver PROD_C5 for receiving a moving image, and an image processor PROD_C6 for generating or modifying an image. In part (a) of Fig. 28, all of them are included in the recording apparatus PROD_C, by way of example. However, some of them may be omitted.

[0838]

The receiver PROD_C5 may be configured to receive a moving image that has not been encoded, or may be configured to receive encoded data that has been encoded using a transmission coding scheme different from a recording coding scheme. In the latter case, a transmission decoder (not illustrated) may be disposed between the receiver PROD_C5 and the encoder PROD_C1 to decode encoded data encoded using a transmission coding scheme.

[0839]

Examples of the recording apparatus PROD_C include a DVD recorder, a BD recorder, and an HDD (Hard Disk Drive) recorder (in this case, the input terminal PROD_C4 or the receiver PROD_C5 serve as a main source from which a moving image is supplied). Other examples of the recording apparatus PROD_C include a camcorder (in this case, the camera PROD_C3 serves as a main source from which a moving image is supplied), a personal computer (in this case, the receiver PROD_C5 or the image processor C6 serves as a main source from which a moving image is supplied), and a smartphone (in this case, the camera PROD_C3 or the receiver PROD_C5 serves as a main source from which a moving image is supplied).

[0840]

Part (b) of Fig. 28 is a block diagram illustrating a configuration of a reproducing apparatus PROD_D including the video decoding device 1 described above. As illustrated in part (b) of Fig. 28, the reproducing apparatus PROD_D includes a reader PROD_D1 for reading encoded data written in a recording medium PROD_M, and a decoder PROD_D2 for decoding the encoded data read by the reader PROD_D1 to obtain a moving image. The video decoding device 1 described above may be used as the decoder PROD_D2.

[0841]

The recording medium PROD_M may be (1) of a type incorporated in the reproducing apparatus PROD_D, such as an HDD or an SSD, or (2) of a type connected to the reproducing apparatus PROD_D, such as an SD memory card or a USB flash

Amendment (CLAIMS)

[Claim 1]

An image decoding device for decoding an image in a prediction unit using, as an inter-frame prediction scheme, a uni-prediction scheme in which one reference image is referred to or a bi-prediction scheme in which two reference images are referred to, the image decoding device comprising:

a motion compensation parameter derivation unit configured to derive a motion compensation parameter indicating one of the uni-prediction scheme and the bi-prediction scheme, wherein

in a case where the prediction unit has a size less than or equal to a certain value,

the motion compensation parameter derivation unit derives the motion compensation parameter by switching the prediction scheme to the uni-prediction scheme.

[Claim 2]

The image decoding device according to Claim 1, wherein

the motion compensation parameter at least includes a first prediction list utilization flag indicating whether or not a first reference prediction list is to be used, and a second prediction list utilization flag indicating whether or not a second reference prediction list is to be used, and

the motion compensation parameter derivation unit derives the motion compensation parameter using the first prediction list utilization flag and the second prediction list utilization flag.

[Claim 3]

The image decoding device according to Claim 2, wherein

in a case where the first prediction list utilization flag indicates that the first prediction list is to be used and the second prediction list utilization flag indicates that the second prediction list is to be used,

the motion compensation parameter derivation unit performs transformation so that one of the first prediction list utilization flag and the second prediction list utilization flag is not used.

[Claim 4]

The image decoding device according to Claim 3, wherein the size of the prediction unit is calculated using a width and height of the prediction unit.

[Claim 5]

An image decoding method for decoding an image in a prediction unit using, as an inter-frame prediction scheme, a uni-prediction scheme in which one reference image is referred to or a bi-prediction scheme in which two reference images are referred to, the image decoding method comprising the steps of:

deriving a motion compensation parameter indicating one of the uni-prediction scheme and the bi-prediction scheme; and

determining whether or not the prediction unit has a size less than or equal to a certain value, wherein

the step of deriving a motion compensation parameter includes deriving the motion compensation parameter by switching the prediction scheme to the uni-prediction scheme in a case where the size of the prediction unit is less than or equal to the certain value.

Dated this 01 day of April 2014

(Arindam Paul)
Reg. No.: IN/PA – 174
Of De Penning & De Penning
Agent for the Applicants

Clean

of PU partition types. Parts (a) to (h) of Fig. 4 illustrate partition shapes in the PU partition types of 2N×2N, 2N×N, 2N×nU, 2N×nD, N×2N, nL×2N, nR×2N and N×N, respectively.

[Fig. 5] Fig. 5 is a diagram illustrating a specific example configuration of a PU size table in which the numbers of PUs and PU sizes are defined in association with CU sizes and PU partition types.

[Fig. 6] Fig. 6 is a diagram illustrating a 2N×N CU and a 2N×nU CU in which an edge having an inclination is present.

[Fig. 7] Fig. 7 is a table illustrating an example of binarization information that defines associations between combinations of CU prediction types and PU partition types and bin sequences.

[Fig. 8] Fig. 8 is a diagram illustrating an example of the binarization information that defines a CU having an 8×8 size.

[Fig. 9] Fig. 9 is a diagram illustrating another example of the binarization information that defines a CU having an 8×8 size.

[Fig. 10] Fig. 10 is a table illustrating another example of the binarization information that defines associations between combinations of CU prediction types and PU partition types and bin sequences.

[Fig. 11] Fig. 11 is a table illustrating another example of the binarization information that defines prediction restriction on a basic inter PU, bi-prediction restriction on a merge PU, skipped derivation of bi-predictive merge candidates are uniformly applied to PUs having 4×4, 4×8, and 8×4 sizes, parts (b) and (c) are diagrams illustrating an example in which the restriction of bi-prediction is imposed only on a basic inter PU without the application of the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates, and part (d) is a diagram illustrating an example in which the bi-prediction restriction on the basic inter PU is uniformly applied to PUs having 4×4, 4×8, and 8×4 sizes and in which the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates are applied to PUs having an 8×8 size.

[Fig. 53] Fig. 53 includes diagrams depicting an example of bi-prediction restriction methods, in which part (a) is a diagram illustrating an example in which the bi-prediction restriction on the basic inter PU and the skipped derivation of bi-predictive merge candidates are applied to 4×4, 4×8, 8×4, and 8×8, and part (b) is a diagram illustrating an example in which the skipped derivation of bi-predictive merge candidates is applied to 4×4, 4×8, 4×8, and 8×8.

[Fig. 54] Fig. 54 is a block diagram illustrating a configuration of a basic motion compensation parameter included in the target prediction tree.

[0121]

The CU information decoding unit 11 supplies the PT information PTI obtained for the target CU to the PU information decoding unit 12. Further, the CU information decoding unit 11 supplies the TT information TTI obtained for the target CU to the TU information decoding unit 13.

[0122]

[PU information decoding unit]

The PU information decoding unit 12 performs a decoding process on the PT information PTI supplied from the CU information decoding unit 11, using the decoding module 10 on the PU level. Specifically, the PU information decoding unit 12 decodes the PT information PTI using the following procedure.

[0123]

The PU information decoding unit 12 refers to the PU partition type information PartMode, and determines the PU partition type for the target prediction tree. Then, the PU information decoding unit 12 sequentially designates each of the PUs included in the target prediction tree as a target PU, and executes the decoding process on the PU information corresponding to the target PU.

[0124]

That is, the PU information decoding unit 12 decodes is reduced and throughput is increased. In addition, a memory for accumulating probabilities of occurrence (states) corresponding to contexts is not necessary. Coding with a fixed probability of 0.5 may be referred to as EP coding (equal probabilities, equal probability coding) or bypass.

[0171]

The operations and effects of the configuration described above will be described with reference to Fig. 6. A context is effective for the improvement in coding efficiency when the same code appears consecutively in a specific condition. Coding efficiency is improved by decoding the suffix portion by referring to contexts when, specifically, 2N×N, 2N×nU, or 2N×nD is consecutively selected in a state where a landscape-oriented partition is selected. This effect works, for example, when 2N×N is selected in a prediction unit subsequent to the prediction unit in which 2N×N was selected.

[0172]

On the other hand, partitions are generally set so as not to lie over edge boundaries, as illustrated in Fig. 6.

[0173]

Specifically, as illustrated in Fig. 6, if an edge E1 having an inclination is present in a region, the PU partition type of a CU 10 and a CU 20 is determined so that no partitions lie across the edge E1.

In the association table BT1, two CU prediction types, namely, the intra CU described above (labeled as "Intra") and inter CU (labeled as "Inter"), are defined for the definition of the respective CU sizes. PU partition types are further defined for the respective CU prediction types.

[0192]

Details are as follows. First, for the intra CU, two PU partition types, namely, 2N×2N and N×N, are defined.

[0193]

A description of 2N×2N will be given hereinafter. In the non-8×8 CU 1012B, only the prefix portion is defined and the bin sequence is "000". The suffix portion is not coded. In the 8×8 CU 1012A, the prefix portion is "000" and the suffix portion is "0".

[0194]

For N×N, on the other hand, a definition is provided only for the non-8×8 CU 1012B. In this case, the prefix portion is "000", and the suffix portion is "1".

[0195]

In this manner, for the intra CU, the prefix portion is "000", which is common.

[0196]

For the inter CU, seven PU partition types, namely, 2N×2N, 2N×N, 2N×nU, 2N×nD, N×2N, nL×2N, and nR×2N, are defined.

[0341]

Part (b) of Fig. 33 illustrates an example of a syntax table in a case that the inter prediction flag is a ternary flag. If a combined list is used, two types, namely, Pred_LC, which means uni-prediction in which one reference frame in an LC list is used, and Pred_Bi, which means bi-prediction, are identified from each other by inter_pred_flag. Otherwise, three types, namely, Pred_L0, which means uni-prediction with the L0 list, Pred_L1, which means uni-prediction with the L1 list, and Pred_Bi, which means bi-prediction, are identified from one another. If the slice is a B slice and bi-prediction is active (DisableBiPred = false), the encoded data includes a first inter prediction flag inter_pred_flag0 for specifying uni-prediction and bi-prediction. If bi-prediction is not active, only in a case that a combined list is not used, the encoded data includes a second inter prediction flag inter_pred_flag1 for specifying uni-prediction and bi-prediction to specify a reference list. The case that a combined list is not used is determined specifically using !UsePredRefLC && !NoBackPredFlag, as illustrated in part (a) of Fig. 33. That is, the determination is based on a flag UsePredRefLC (indicating that a combined list is used if the value of UsePredRefLC is true) specifying whether or not to use a combined list, and a flag NoBackPredFlag (indicating that backward prediction is not used if the value of NoBackPredFlag is true) specifying whether or not to use backward prediction. If a combined list is used, the use of a combined list is determined without list selection. No use of backward prediction means the disabling of Pred_L1. In this case, it may be determined that the list used also when the second inter prediction flag inter_pred_flag1 is not encoded is the combined list (Pred_LC) or the L0 list (Pred_L1). The expression "NoL1PredFlag", which means no use of the L1 list, may be used instead of NoBackPredFlag.

[0342]

A threshold value used to determine whether or not to impose the restriction of bi-prediction or to determine the PU size in a case that the restriction of bi-prediction is imposed may be included in the encoded data. Fig. 34 illustrates an example of a syntax table for bi-prediction restriction. Part (a) of Fig. 34 illustrates the case that the sequence parameter set includes the flag disable_bipred_in_small_PU restricting whether or not to impose the restriction of bi-prediction. As illustrated in Part (a) of Fig. 34, a flag for the restriction of bi-prediction may be encoded independently from a flag disable_inter_4x4 prohibiting a small size PU (here, a 4×4 size PU). The purpose of the flag prohibiting a small size PU is also to reduce the amount of worst-case processing to

1212J. Although not illustrated in Fig. 43, the neighboring merge candidate derivation unit 1212A and the temporal merge candidate derivation unit 1212B are supplied with decoding parameters for an already decoded CU and PU, which are stored in the frame memory 16, particularly, motion compensation parameters on a per-PU basis. In the following, the neighboring merge candidate derivation unit 1212A, the temporal merge candidate derivation unit 1212B, the unique candidate derivation unit 1212C, the combined bi-predictive merge candidate derivation unit 1212D, the non-scaled bi-predictive merge candidate derivation unit 1212E, and the zero vector merge candidate derivation unit 1212F are collectively referred to as "merge candidate deriving means".

[0359]

In the merge motion compensation parameter derivation unit 1212, the merge candidate derivation control unit 1212G controls each merge candidate deriving means to derive a predetermined number MRG_MAX_NUM_CANDS of merge candidates, and stores the derived merge candidates in the merge candidate storage unit 1212H. Here, each merge candidate is composed of prediction list utilization flags predFlagL0 and predFlagL1, which are motion compensation parameters of a PU, reference index numbers refIdxL0 and refIdxL1, and motion vectors mvL0 and mvL1. The merge candidate storage unit 1212H stores sets of the motion compensation parameters described above reference picture (S102). In S103, a redundant merge candidate is removed from among the derived merge candidates A0 to T, and the remaining merge candidates are stored in the merge candidate storage unit 1212H. If the number of non-redundant merge candidates is greater than or equal to MRG_MAX_NUM_CANDS, the derivation of merge candidates is terminated (YES in S104). Otherwise (NO in S104), the operation proceeds to S105. In the case of a B slice (YES in S105), the operation proceeds to S106, and otherwise (NO in S105), the operation skips S106, S107 and S108 and proceeds to S109 (S105). Also in a case that the restriction of bi-prediction is imposed, here, in the case of a small size PU, which corresponds to the case that the bi-directional derivation of merge candidates is skipped, the operation skips the bi-predictive motion candidate derivation process of S107 and S108, and proceeds to S109 (S106). In S107, the combined bi-predictive merge candidate derivation unit 1212D derives combined bi-predictive merge candidates, and the derived combined bi-predictive merge candidates are stored in the merge candidate storage unit 1212H. In S108, the non-scaled bi-predictive merge candidate derivation unit 1212E derives non-scaled bi-predictive merge candidates, and the derived non-scaled bi-predictive merge candidates are stored in the merge candidate storage unit 1212H. Here, if the number of merge candidates is greater than or equal to plurality of merge candidates. This similarly applied to the other merge candidate derivation units described hereinafter.

[0364]

In a case that a neighboring block is not available (unavailable) or in the case of an intra block, the corresponding merge candidates are not derived. A neighboring block is not available in a case that the block is located outside the screen, the block is located outside the slice, or the block is an undecoded block in the scan order of blocks. The positions A0 to B2 may be expressed as follows, where the upper left coordinates of a PU is represented by (xP, yP) and the PU has sizes nPSW and nPSH.

A0: (xP - 1, yP + nPSH)

A1: (xP - 1, yP + nPSH - 1)

B0: (xP + nPSH, yP - 1)

B1: (xP + nPSH - 1, yP - 1)

B2: (xP - 1, yP - 1)

If all the merge candidates corresponding to the positions A0, A1, B0, and B1 are successfully derived, the merge candidate corresponding to the position B2 is not derived. In the derivation of merge candidates for the PU partition type of 2N×N or N×2N and the PU index of 1, the following operation is performed. Only in a case that the motion compensation parameters for each merge candidate do refIdxL0 that occupies substantially the same spatial position as the spatial position of the target PU in the current picture or by copying motion compensation parameters of a reference picture PU specified by the reference index number refIdxL0. The method for deriving the reference index number refIdxL0 and the reference index number refIdxL1 will be described with reference to part (b) of Fig. 46. The reference index number refIdxLX (where X is 0, 1, or C) is determined using the reference pictures refIdxLXA, refIdxLXB, and refIdxLXC of neighboring PUs A, B, and C of the target PU as follows.

(1) In the case of refIdxLXA = refIdxLXB = refIdxLXC,
if refIdxLXA = -1, refIdxLX = 0
otherwise, refIdxLX = refIdxLXA

(2) In the case of refIdxLXA = refIdxLXB,
if refIdxLXA = -1, refIdxLX = refIdxLXC
otherwise, refIdxLX = refIdxLXA

(3) In the case of refIdxLXB = refIdxLXC,
if refIdxLXB = -1, refIdxLX = refIdxLXA
otherwise, refIdxLX = refIdxLXB

(4) In the case of refIdxLXA = refIdxLXC,
if refIdxLXA = -1, refIdxLX = refIdxLXB
otherwise, refIdxLX = refIdxLXA

(5) In the case of refIdxLXA = -1,
refIdxLX = min (refIdxLXB, refIdxLXC)

derivation unit 1213B, a zero vector merge candidate derivation unit 1213F, a motion vector candidate derivation control unit 1213G, a motion vector candidate storage unit 1213H, a motion vector candidate selection unit 1213I, and a motion vector restoration unit 1213J. In the following, the neighboring motion vector candidate derivation unit 1213A, the temporal motion vector candidate derivation unit 1213B, and the zero vector merge candidate derivation unit 1213F are collectively referred to as "motion vector/merge candidate deriving means".

[0376]

In the basic motion compensation parameter derivation unit 1213, the motion vector candidate derivation control unit 1213G controls each motion vector/merge candidate deriving means to derive a predetermined number PMV_MAX_NUM_CANDS of predictive motion vector candidates, and stores the derived predictive motion vector candidates in the motion vector candidate storage unit 1213H. Here, each predictive motion vector candidate is composed of motion vectors mvL0 and mvL1. The motion vector candidate storage unit 1213H stores the combinations of motion compensation parameters described above as predictive motion vector candidates. The stored predictive motion vector candidates are managed as lists (predictive motion vector candidate lists) in which the predictive motion vector candidates are operation without decoding the inter prediction flag inter_pred_flag. If the PU size is a size other than the small PU size (if DisableBiPred != true) (NO in S142), the inter prediction flag decoding unit 1028 decodes the combined inter prediction reference index combined_inter_pred_ref_idx through S143, S144, and S145. If the PU size is the small PU size (YES in S142), the inter prediction flag decoding unit 1028 decodes the combined inter prediction reference index combined_inter_pred_ref_idx through S146, S147, and S148.

[0409]

In S143 and S146, a maximum value MaxPredRef is calculated. The maximum value MaxPredRef is as illustrated in the table TBL37 and the pseudo code CODE37 in part (d) of Fig. 37. Specifically, the maximum value MaxPredRef for a size other than the small size PU, that is, for no restriction of bi-prediction (DisableBiPred != true), is calculated by NumPredRefLC+NumPredRefL0*NumPredRefL1 or NumPredRefL0+NumPredRefL0*NumPredRefL1, that is, by determining the sum of the number of uni-predictive combined reference picture sets (NumPredRefLC or NumPredRefL0) and the number of bi-predictive combined reference picture sets (NumPredRefL0*NumPredRefL1) (S143). The maximum value MaxPredRef for restriction of bi-prediction (DisableBiPred = true) is calculated by NumPredRefLC or NumPredRefL0, that is, encoding process of combined_inter_pred_ref_idx in a case that a variable table is used.

[0412]

In the foregoing, a description has been given of the following methods for reducing the amount of processing for a small PU size: bi-prediction restriction on the basic inter PU (change in the method for decoding the inter prediction flag and the combined inter prediction reference index), bi-prediction restriction on the merge PU (bi-prediction/uni-prediction conversion in the derivation of merge candidates), and skipped calculation of bi-predictive merge candidates. These restrictions may be used individually, or PU sizes to be subject to these restrictions may have different values. Fig. 52 and Fig. 53 illustrate an example of the reduction in the amount of processing for bi-prediction. In Fig. 52 and Fig. 53, open circles indicate that the processes are performed, and crosses indicate that the processes are not performed.

[0413]

Part (a) of Fig. 52 illustrates an example in which the bi-prediction restriction on the basic inter PU, the bi-prediction restriction on the merge PU, and the skipped derivation of bi-predictive merge candidates are uniformly applied to the PUs having the sizes of 4×4, 4×8, and 8×4. Parts (b) and (c) of Fig. 52 illustrate an example in which the restriction of bi-prediction is imposed only on the basic inter PU without the application of the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates. In general, the restriction of bi-prediction for the merge PU may cause a reduction in coding efficiency. It is thus appropriate that the restriction of bi-prediction is imposed only on the basic inter PU.

[0414]

Part (d) of Fig. 52 illustrates an example in which the bi-prediction restriction on the basic inter PU is uniformly applied to the PUs having sizes of 4×4, 4×8, and 8×4 and in which the bi-prediction restriction on the merge PU and the skipped derivation of bi-predictive merge candidates are applied to the PUs having an 8×8 size. Relaxing the bi-prediction restriction on the merge PU compared to the bi-prediction restriction on the basic inter PU will be appropriate in terms of coding efficiency.

[0415]

Part (a) of Fig. 53 illustrates an example in which the bi-prediction restriction on the basic inter PU and the skipped derivation of bi-predictive merge candidates are applied to 4×4, 4×8, 8×4, and 8×8. Simplifying the derivation of merge candidates used as motion compensation parameters of a merge PU, without the restriction of bi-prediction on the merge PU, may reduce the amount of processing regarding bi-prediction in the merge PU. Part (b) of Fig. 53 illustrates an example in which the skipped derivation of bi-predictive merge candidates is applied to 4×4, 4×8, 8×4, and 8×8. In this manner, the skipped derivation of bi-predictive merge candidates may be used alone.

[0416]

In order to implement the cases described above, determination methods may be managed using different flags. For example, flags DisableBiPredFlag, DisableBiPredMerge, and DisableBiPredMergeDerive for imposing the respective prediction restrictions are provided, and are made feasible through the following operation.

[0417]

For example, the bi-prediction restricted PU determination unit 1218 individually derives three flags DisableBiPredFlag, DisableBiPredMerge, and DisableBiPredMergeDerive. In the example illustrated in part (d) of Fig. 52, the flags may be derived as follows.

[0418]

DisableBiPredFlag = (log2CUSize == 3 && PU partition type != 2Nx2N) ? true : false

DisableBiPredMerge, DisableBiPredMergeDerive = ((log2CUSize == 4 && PU partition type == NxN) || log2CUSize

size) is 3, and is not particularly restricted, whereas the minimum PU size is 8×4 or 4×8. That is, the use of 4×4 PU is disabled. In addition, the minimum bi-prediction PU size is 8×4 or 4×8, and the use of bi-prediction for 4×4 PU is disabled. If the level level_idc is greater than or equal to the predetermined threshold value TH2, the logarithmic value of the minimum CU size is 4, and the minimum PU size is restricted to 8×8. That is, the use of 8×4 PU, 4×8 PU, and 4×4 PU is disabled. In addition, the minimum bi-prediction PU size is 16×8, and the use of bi-prediction for 8×8 PU is disabled. The use of bi-prediction for 8×4 PU, 4×8 PU, and 4×4 PU is also disabled due to the restriction of the minimum PU size. It is appropriate that the threshold value TH1 is level 3.1, which is on the line of 720P, and the threshold value TH2 is level 5, which is on the line equivalent to 2560×1600. However, other threshold values may be used.

[0446]

Fig. 85 illustrates another example of level limits of the present invention. The illustrated example is substantially the same as the example in Fig. 84, except that if the level level_idc is less than a predetermined threshold value TH0, the logarithmic value of the minimum CU size, the minimum PU size, and the minimum bi-prediction PU size are 3, 4×4, and 4×4, respectively, and no restrictions restrict_bipred_flag may take one of 0, 1, and otherwise (for example, 2). The following is a description of the individual steps S in the pseudo code illustrated in Fig. 68.

[0506]

S681: If restrict_bipred_flag is equal to "0", the bi-prediction restricted PU determination unit 1218A sets "0" in the variable DisableBipred.

[0507]

S682: If restrict_bipred_flag is equal to "1", the bi-prediction restricted PU determination unit 1218A sets the variable DisableBipred as follows.

[0508]

The restriction of bi-prediction is performed "in a case that the logarithmic value (log2CUSize) of the CU size matches Log2MinCUSize and the PU mode is other than 2N×2N" or "in a case that the logarithmic value (log2CUSize) of the CU size is smaller than Log2MinCUSize".

[0509]

S682: If restrict_bipred_flag is equal to a value other than the values described above (for example, "2"), the bi-prediction restricted PU determination unit 1218A sets the variable DisableBipred as follows.

imposed by the bi-/uni-prediction conversion process is reduced, compared to the case that the bi-/uni-prediction conversion process is performed on all the merge candidates. Furthermore, a configuration in which bi-/uni-prediction conversion is performed on at least one temporal merge candidate ensures that, even if the restriction of bi-prediction is imposed on 8×8 PU having a relatively large PU size, temporal merge candidates for 8×8 PU are uni-predictive, or available. Accordingly, the reduction in coding efficiency may be minimized. In addition, the merge candidate derivation process and the bi-/uni-prediction conversion process are executed in parallel, resulting in processing being efficiently performed.

[0554]

(Modifications)

Preferred modifications 1 to 3 of the present configuration will be described.

[0555]

(Modification 1)

The bi-prediction/uni-prediction conversion unit 1219A may be configured to perform a bi-prediction/uni-prediction conversion process after all the merge candidates have been derived and a merge candidate list has been stored in the merge candidate storage unit 1212H.

[0556]

As already discussed, the configuration illustrated in Fig. 61 and Fig. 71 is a configuration in which the merge candidate derivation process and the bi-/uni-prediction conversion process are executed in parallel.

[0568]

A timetable of the series of processes is as illustrated in, for example, Fig. 74. In a time chart illustrated in Fig. 74, merge candidates A to E have been derived. Among them, two merge candidates are subjected to bi-/uni-prediction conversion by bi-prediction restriction. It is assumed that the time taken to derive the merge candidates C and E is longer than the time taken to derive the merge candidates A, B, and D.

[0569]

As illustrated in Fig. 74, the merge candidates A and B are the first two merge candidates among the merge candidates A to E, and are thus targets of bi-/uni-prediction conversion. The bi-/uni-prediction conversion of the merge candidates A and B is performed in parallel to the execution of the merge candidates C and E, which may require a longer processing time. In the example illustrated in Fig. 74, thus, a list creation process is started upon the completion of the derivation process for the merge candidates C and E, and the entire process ends when the list creation process is completed.

integers will be disclosed with reference to Fig. 76. In the configuration illustrated in Fig. 76, the bi-prediction/uni-prediction conversion unit 1219A in the configuration illustrated in Fig. 61 is replaced by a motion-vector-to-integer conversion unit 1220.

[0590]

The motion-vector-to-integer conversion unit 1220 converts at least one component among one or more non-integer components included in non-integer motion vectors into an integer component. The conversion performed by the motion-vector-to-integer conversion unit 1220 is hereinafter referred to as the conversion of motion vectors into integers.

[0591]

More specifically, in a case that the restriction of bi-prediction is to be imposed, if the merge candidates input from the neighboring merge candidate derivation unit 1212A or the temporal merge candidate derivation unit 1212B include bi-predictive merge candidates, the motion-vector-to-integer conversion unit 1220 determines whether or not the two motion vectors of the bi-predictive merge candidates are non-integer motion vectors. If at least one of the two motion vectors of the bi-predictive merge candidates is a non-integer motion vector, the motion-vector-to-integer conversion unit 1220 converts the non-integer motion vector configured to encode an input image #10 to generate encoded data #1, and to output the encoded data #1.

[0722]

(Configuration of video encoding device)

First, an example configuration of the video encoding device 2 will be described with reference to Fig. 25. Fig. 25 is a functional block diagram illustrating a configuration of the video encoding device 2. As illustrated in Fig. 25, the video encoding device 2 includes an encoding setting unit 21, a dequantization/inverse transform unit 22, a prediction image generation unit 23, an adder 24, a frame memory 25, a subtractor 26, a transform/quantization unit 27, an encoded data generation unit (encoding means) 29, and a PU information generation unit 30.

[0723]

The encoding setting unit 21 generates image data and various kinds of setting information concerning encoding, on the basis of the input image #10.

[0724]

Specifically, the encoding setting unit 21 generates the following image data and setting information.

[0725]

First, the encoding setting unit 21 sequentially splits the input image #10 into slices and tree blocks to generate a CU image #100 for a target CU.

The subtractor 26 subtracts the prediction image Pred from the CU image #100 to generate a prediction residual D for the target CU. The subtractor 26 supplies the generated prediction residual D to the transform/quantization unit 27.

[0736]

The transform/quantization unit 27 applies orthogonal transform and quantization to the prediction residual D to generate a quantized prediction residual. The term "orthogonal transform", as used herein, refers to an orthogonal transform from the pixel domain to the frequency domain. Examples of the orthogonal transform include DCT transform (Discrete Cosine Transform) and DST transform (Discrete Sine Transform).

[0737]

Specifically, the transform/quantization unit 27 refers to the CU image #100 and the CU information CU', and determines the pattern in which the target CU is split into one or a plurality of blocks. The transform/quantization unit 27 splits the prediction residual D into prediction residuals for the respective blocks in accordance with the determined partition pattern.

[0738]

Further, the transform/quantization unit 27 applies orthogonal transform to the prediction residual for each block to generate a prediction residual in the frequency domain. Then, the transform/quantization unit 27 quantizes the prediction residual in the frequency domain to generate a quantized prediction residual for each block.

[0739]

The transform/quantization unit 27 further generates TT setting information TTI' including the generated quantized prediction residual for each block, TT split information for specifying the partition pattern of the target CU, and information concerning all possible patterns of splitting the target CU into individual blocks. The transform/quantization unit 27 supplies the generated TT setting information TTI' to the dequantization/inverse transform unit 22 and the encoded data generation unit 29.

[0740]

The PU information generation unit 30 encodes the PT setting information PTI' if the prediction type indicated by the PT setting information PTI' is inter prediction, and derives PT setting information PTI. The PU information generation unit 30 further generates PTI setting information PTI' on merge candidates, and supplies the PTI setting information PTI' to the encoding setting unit 21.

[0741]

The encoded data generation unit 29 encodes the header information H', the TT setting information TTI', and the PT setting information PTI'. The encoded data generation unit compensation parameter derivation unit 1212", and "the reference frame setting information storage unit 123" in the description of the example configuration [2-3-3] should be substituted with "the motion compensation parameter generation unit 301", "the merge motion compensation parameter generation unit 3012", and "the configuration corresponding to the reference frame setting information storage unit 123", respectively.

[0788]

[2-3-4]' Conversion of motion vectors into integers

The PU information generation unit 30 may convert motion vectors into integers. A specific configuration is as illustrated in Fig. 83. In the configuration illustrated in Fig. 83, the bi-prediction/uni-prediction conversion unit 1219 in the PU information generation unit 30 illustrated in Fig. 55 or 56 is replaced by a bi-prediction/uni-prediction conversion unit 1219A. In the configuration illustrated in Fig. 83, the bi-prediction/uni-prediction conversion unit 1219 in the PU information generation unit 30 illustrated in Fig. 55 or 56 is replaced by a motion-vector-to-integer conversion unit 1220.

[0789]

More details are similar to those described in, for example, the description of the example configuration [2-3-4] of the video decoding device 1, and a description thereof

[0815]

(Processing flow)

The CU encoding process of the video encoding device 2 will be described hereinafter with reference to Fig. 26. In the following, it is assumed that a target CU is an inter CU or a skip CU. Fig. 26 is a flowchart illustrating an example of the flow of the CU encoding process (inter/skip CU) of the video encoding device 2.

[0816]

When the CU encoding process starts, the encoding setting unit 21 determines CU prediction information on the target CU, and the encoded data generation unit 29 encodes the CU prediction information determined by the encoding setting unit 21 (S21). This process is performed on a per-CU basis.

[0817]

Specifically, the encoding setting unit 21 determines whether or not the target CU is a skip CU. If the target CU is a skip CU, the encoding setting unit 21 causes the encoded data generation unit 29 to encode the skip flag SKIP. If the target CU is not a skip CU, the encoding setting unit 21 causes the encoded data generation unit 29 to encode the CU prediction type information Pred_type.

[0818]

Then, processing is performed on a per-PU basis.

Disc: registered trademark).

[0837]

The recording apparatus PROD_C may further include sources from which moving images to be input to the encoder PROD_C1 are supplied, including a camera PROD_C3 for capturing a moving image, an input terminal PROD_C4 through which a moving image is input from outside, a receiver PROD_C5 for receiving a moving image, and an image processor PROD_C6 for generating or modifying an image. In part (a) of Fig. 28, all of them are included in the recording apparatus PROD_C, by way of example. However, some of them may be omitted.

[0838]

The receiver PROD_C5 may be configured to receive a moving image that has not been encoded, or may be configured to receive encoded data that has been encoded using a transmission coding scheme different from a recording coding scheme. In the latter case, a transmission decoder (not illustrated) may be disposed between the receiver PROD_C5 and the encoder PROD_C1 to decode encoded data encoded using a transmission coding scheme.

[0839]

Examples of the recording apparatus PROD_C include a DVD recorder, a BD recorder, and an HDD (Hard Disk Drive) recorder (in this case, the input terminal PROD_C4 or the receiver PROD_C5 serve as a main source from which a moving image is supplied). Other examples of the recording apparatus PROD_C include a camcorder (in this case, the camera PROD_C3 serves as a main source from which a moving image is supplied), a personal computer (in this case, the receiver PROD_C5 or the image processor C6 serves as a main source from which a moving image is supplied), and a smartphone (in this case, the camera PROD_C3 or the receiver PROD_C5 serves as a main source from which a moving image is supplied).

[0840]

Part (b) of Fig. 28 is a block diagram illustrating a configuration of a reproducing apparatus PROD_D including the video decoding device 1 described above. As illustrated in part (b) of Fig. 28, the reproducing apparatus PROD_D includes a reader PROD_D1 for reading encoded data written in a recording medium PROD_M, and a decoder PROD_D2 for decoding the encoded data read by the reader PROD_D1 to obtain a moving image. The video decoding device 1 described above may be used as the decoder PROD_D2.

[0841]

The recording medium PROD_M may be (1) of a type incorporated in the reproducing apparatus PROD_D, such as an HDD or an SSD, or (2) of a type connected to the reproducing apparatus PROD_D, such as an SD memory card or a USB flash

Amendment (CLAIMS)

[Claim 1]

An image decoding device for decoding an image in a prediction unit using, as an inter-frame prediction scheme, a uni-prediction scheme in which one reference image is referred to or a bi-prediction scheme in which two reference images are referred to, the image decoding device comprising:

a motion compensation parameter derivation unit configured to derive a motion compensation parameter indicating one of the uni-prediction scheme and the bi-prediction scheme, wherein

in a case where the prediction unit has a size less than or equal to a certain value,

the motion compensation parameter derivation unit derives the motion compensation parameter by switching the prediction scheme to the uni-prediction scheme.

[Claim 2]

The image decoding device according to Claim 1, wherein

the motion compensation parameter at least includes a first prediction list utilization flag indicating whether or not a first reference prediction list is to be used, and a second prediction list utilization flag indicating whether or not a second reference prediction list is to be used, and

the motion compensation parameter derivation unit derives the motion compensation parameter using the first prediction list utilization flag and the second prediction list utilization flag.

[Claim 3]

The image decoding device according to Claim 2, wherein

in a case where the first prediction list utilization flag indicates that the first prediction list is to be used and the second prediction list utilization flag indicates that the second prediction list is to be used,

the motion compensation parameter derivation unit performs transformation so that one of the first prediction list utilization flag and the second prediction list utilization flag is not used.

[Claim 4]

The image decoding device according to Claim 3, wherein the size of the prediction unit is calculated using a width and height of the prediction unit.

[Claim 5]

An image decoding method for decoding an image in a prediction unit using, as an inter-frame prediction scheme, a uni-prediction scheme in which one reference image is referred to or a bi-prediction scheme in which two reference images are referred to, the image decoding method comprising the steps of:

deriving a motion compensation parameter indicating one of the uni-prediction scheme and the bi-prediction scheme; and

determining whether or not the prediction unit has a size less than or equal to a certain value, wherein

the step of deriving a motion compensation parameter includes deriving the motion compensation parameter by switching the prediction scheme to the uni-prediction scheme in a case where the size of the prediction unit is less than or equal to the certain value.

Dated this 01 day of April 2014

(Arindam Paul)
Reg. No.: IN/PA – 174
Of De Penning & De Penning
Agent for the Applicants