(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 February 2019 (28.02.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/039322 A1

(51) International Patent Classification:
H04N 19/537 (2014.01) *H04N 19/157* (2014.01)
H04N 19/105 (2014.01) *H04N 19/96* (2014.01)

(21) International Application Number: PCT/JP2018/030059

(22) International Filing Date: 10 August 2018 (10.08.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/548,631 22 August 2017 (22.08.2017) US
62/698,785 16 July 2018 (16.07.2018) US

(71) Applicant: **PANASONIC INTELLECTUAL PROPERTY CORPORATION OF AMERICA** [US/US]; 20000 Mariner Avenue, Suite 200, Torrance, California, 90503 (US).

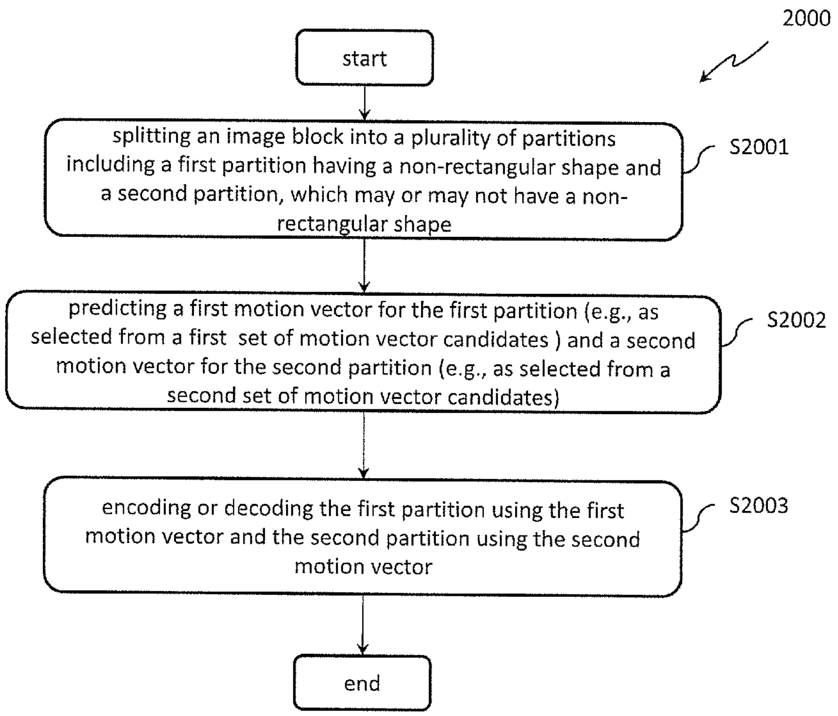
(72) Inventors: **ABE, Kiyofumi**; c/o Panasonic Corporation, 1006, Oaza Kadoma, Kadoma-shi, Osaka, 5718501 (JP). **NISHI, Takahiro**. **TOMA, Tadamasa**. **KANOH, Ryuichi**. **LIM, Chong Soon**. **LIAO, Ru Ling**. **SUN, Hai Wei**. **SHASHIDHAR, Sughosh Pavan**. **TEO, Han Boon**. **LI, Jing Ya**.

(74) Agent: **NII, Hiromori** et al.; c/o NII Patent Firm, 6F, Tanaka Ito Pia Shin-Osaka Bldg., 3-10, Nishi Nakajima 5-chome, Yodogawa-ku, Osaka-city, Osaka, 5320011 (JP).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

(54) **Title:** IMAGE ENCODER, IMAGE DECODER, IMAGE ENCODING METHOD, AND IMAGE DECODING METHOD



(57) **Abstract:** An image encoder is provided, which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape (e.g., a triangular shape) and a second partition; predicting a first motion vector for the first partition and a second motion vector for the second partition; and encoding the first partition using the first motion vector and the second partition using the second motion vector.

WO 2019/039322 A1

[Continued on next page]

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

Description

Title of Invention: IMAGE ENCODER, IMAGE DECODER, IMAGE ENCODING METHOD, AND IMAGE DECODING METHOD

Technical Field

[0001] This disclosure relates to video coding, and particularly to video encoding and decoding systems, components, and methods for performing an inter prediction function to build a current block based on a reference frame or an intra prediction function to build a current block based on an encoded/decoded reference block in a current frame.

Background Art

[0002] With advancement in video coding technology, from H.261 and MPEG-1 to H.264/AVC (Advanced Video Coding), MPEG-LA, H.265/HEVC (High Efficiency Video Coding) and H.266/VVC (Versatile Video Codec), there remains a constant need to provide improvements and optimizations to the video coding technology to process an ever-increasing amount of digital video data in various applications. This disclosure relates to further advancements, improvements and optimizations in video coding, particularly, in connection with an inter prediction function or an intra prediction function, splitting an image block into a plurality of partitions including at least a first partition having a non-rectangular shape (e.g., a triangle) and a second partition.

Summary of Invention

[0003] According to one aspect, an image encoder is provided including circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicting a first motion vector for the first partition and a second motion vector for the second partition; and encoding the first partition using the first motion vector and the second partition using the second motion vector.

[0004] Some implementations of embodiments of the present disclosure may improve an encoding efficiency, may simply be an encoding/decoding process, may accelerate an encoding/decoding process speed, may efficiently select appropriate components/operations used in encoding and decoding such as appropriate filter, block size, motion vector, reference picture, reference block, etc.

[0005] Additional benefits and advantages of the disclosed embodiments will become apparent from the specification and drawings. The benefits and/or advantages may be individually obtained by the various embodiments and features of the specification and drawings, not all of which need to be provided in order to obtain one or more of such benefits and/or advantages.

[0006] It should be noted that general or specific embodiments may be implemented as a system, a method, an integrated circuit, a computer program, a storage medium, or any selective combination thereof.

Brief Description of Drawings

[0007] [fig.1]FIG. 1 is a block diagram illustrating a functional configuration of an encoder according to an embodiment.

[fig.2]FIG. 2 illustrates one example of block splitting.

[fig.3]FIG. 3 is a table indicating transform basis functions of various transform types.

[fig.4A]FIG. 4A illustrates one example of a filter shape used in ALF (adaptive loop filter).

[fig.4B]FIG. 4B illustrates another example of a filter shape used in ALF.

[fig.4C]FIG. 4C illustrates another example of a filter shape used in ALF.

[fig.5A]FIG. 5A illustrates 67 intra prediction modes used in an example of intra prediction.

[fig.5B]FIG. 5B is a flow chart illustrating one example of a prediction image correction process performed in OBMC (overlapped block motion compensation) processing.

[fig.5C]FIG. 5C is a conceptual diagram illustrating one example of a prediction image correction process performed in OBMC processing.

[fig.5D]FIG. 5D is a flow chart illustrating one example of FRUC (frame rate up conversion) processing.

[fig.6]FIG. 6 illustrates one example of pattern matching (bilateral matching) between two blocks along a motion trajectory.

[fig.7]FIG. 7 illustrates one example of pattern matching (template matching) between a template in the current picture and a block in a reference picture.

[fig.8]FIG. 8 illustrates a model that assumes uniform linear motion.

[fig.9A]FIG. 9A illustrates one example of deriving a motion vector of each sub-block based on motion vectors of neighboring blocks.

[fig.9B]FIG. 9B illustrates one example of a process for deriving a motion vector in merge mode.

[fig.9C]FIG. 9C is a conceptual diagram illustrating an example of DMVR (dynamic motion vector refreshing) processing.

[fig.9D]FIG. 9D illustrates one example of a prediction image generation method using a luminance correction process performed by LIC (local illumination compensation) processing.

[fig.10]FIG. 10 is a block diagram illustrating a functional configuration of the decoder according to an embodiment.

[fig.11]FIG. 11 is a flowchart illustrating an overall process flow of splitting an image block into a plurality of partitions including at least a first partition having a non-rectangular shape (e.g., a triangle) and a second partition and performing further processing according to one embodiment.

[fig.12]FIG. 12 illustrates two exemplary methods of splitting an image block into a first partition having a non-rectangular shape (e.g., a triangle) and a second partition (also having a non-rectangular shape in the illustrated examples).

[fig.13]FIG. 13 illustrates one example of a boundary smoothing process involving weighting first values of boundary pixels predicted based on the first partition and second values of the boundary pixels predicted based on the second partition.

[fig.14]FIG. 14 illustrates three further samples of a boundary smoothing process involving weighting first values of boundary pixels predicted based on the first partition and second values of the boundary pixels predicted based on the second partition.

[fig.15]FIG. 15 is a table of sample parameters ("first index values") and sets of information respectively encoded by the parameters.

[fig.16]FIG.16 is a table illustrating banalization of parameters (index values).

[fig.17]FIG. 17 is a flowchart illustrating a process of splitting an image block into a plurality of partitions including a first partition having a non-rectangular-shape and a second partition.

[fig.18]FIG. 18 illustrates examples of splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape, which is a triangle in the illustrated examples, and a second partition.

[fig.19]FIG. 19 illustrates further examples of splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape, which is a polygon with at least five sides and angles in the illustrated examples, and a second partition.

[fig.20]FIG. 20 is a flowchart illustrating a boundary smoothing process involving weighting first values of boundary pixels predicted based on the first partition and second values of the boundary pixels predicted based on the second partition.

[fig.21A]FIG. 21A illustrates an example of a boundary smoothing process wherein boundary pixels for which first values to be weighted are predicted based on the first partition and second values to be weighted are predicted based on the second partition.

[fig.21B]FIG. 21B illustrates an example of a boundary smoothing process wherein boundary pixels for which first values to be weighted are predicted based on the first

partition and second values to be weighted are predicted based on the second partition.
[fig.21C]FIG. 21C illustrates an example of a boundary smoothing process wherein boundary pixels for which first values to be weighted are predicted based on the first partition and second values to be weighted are predicted based on the second partition.
[fig.21D]FIG. 21D illustrates an example of a boundary smoothing process wherein boundary pixels for which first values to be weighted are predicted based on the first partition and second values to be weighted are predicted based on the second partition.
[fig.22]FIG. 22 is a flowchart illustrating a method performed on the encoder side of splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, based on a partition parameter indicative of the splitting, and writing one or more parameters including the partition parameter into a bitstream in entropy encoding.
[fig.23]FIG. 23 is a flowchart illustrating a method performed on the decoder side of parsing one or more parameters from a bitstream, which includes a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, and splitting the image block into the plurality of partitions based on the partition parameter, and decoding the first partition and the second partition.
[fig.24]FIG. 24 is a table of sample partition parameters ("first index values") which respectively indicate splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, and sets of information that may be jointly encoded by the partition parameters, respectively.
[fig.25]FIG. 25 is a table of sample combinations of a first parameter and a second parameter, one of which being a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition.
[fig.26]FIG. 26 illustrates an overall configuration of a content providing system for implementing a content distribution service.
[fig.27]FIG. 27 illustrates one example of an encoding structure in scalable encoding.
[fig.28]FIG. 28 illustrates one example of an encoding structure in scalable encoding.
[fig.29]FIG. 29 illustrates an example of a display screen of a web page.
[fig.30]FIG. 30 illustrates an example of a display screen of a web page.
[fig.31]FIG. 31 illustrates one example of a smartphone.
[fig.32]FIG. 32 is a block diagram illustrating a configuration example of a smartphone.

Description of Embodiments

[0008] According to one aspect, an image encoder is provided including circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicting a first motion vector for the first partition and a second motion vector for the second partition; and encoding the first partition using the first motion vector and the second partition using the second motion vector.

[0009] According to a further aspect, the second partition has a non-rectangular shape. According to another aspect, the non-rectangular shape is a triangle. According to a further aspect, the non-rectangular shape is selected from a group consisting of a triangle, a trapezoid, and a polygon with at least five sides and angles.

[0010] According to another aspect, the predicting includes selecting the first motion vector from a first set of motion vector candidates and selecting the second motion vector from a second set of motion vector candidates. For example, the first set of motion vector candidates may include motion vectors of partitions neighboring the first partition, and the second set of motion vector candidates may include motion vectors of partitions neighboring the second partition. The partitions neighboring the first partition and the partitions neighboring the second partition may be outside of the image block from which the first partition and the second partition are split. The neighboring partitions may be one or both of spatially neighboring partitions and temporary neighboring partitions. The first set of motion vector candidates may be the same as, or different from, the second set of motion vector candidates.

[0011] According to another aspect, the predicting includes, selecting a first motion vector candidate from a first set of motion vector candidates and deriving the first motion vector by adding a first motion vector difference to the first motion vector candidate, and selecting a second motion vector candidate from a second set of motion vector candidates and deriving the second motion vector by adding a second motion vector difference to the second motion vector candidate.

[0012] According to another aspect, an image encoder is provided including: a splitter which, in operation, receives and splits an original picture into blocks; an adder which, in operation, receives the blocks from the splitter and predictions from a prediction controller, and subtracts each prediction from its corresponding block to output a residual; a transformer which, in operation, performs a transform on the residuals outputted from the adder to output transform coefficients; a quantizer which, in operation, quantizes the transform coefficients to generate quantized transform coefficients; an entropy encoder which, in operation, encodes the quantized transform coefficients to generate a bitstream; and the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an encoded reference block in a current picture. The prediction controller, in operation, splits the blocks into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicts a first motion vector for the first partition and a second motion vector for the second partition; and encodes the first partition using the first motion vector and the second partition using the second motion vector.

[0013] According to another aspect, an image encoding method is provided, which includes generally three steps: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicting a first motion vector for the first partition and a second motion vector for the second partition; and encoding the first partition using the first motion vector and the second partition using the second motion vector.

[0014] According to another aspect, an image decoder is provided which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicting a first motion vector for the first partition and a second motion vector for the second partition; and decoding the first partition using the first motion vector and the second partition using the second motion vector.

[0015] According to a further aspect, the second partition has a non-rectangular shape. According to another aspect, the non-rectangular shape is a triangle. According to a further aspect, the non-rectangular shape is selected from a group consisting of a triangle, a trapezoid, and a polygon with at least five sides and angles.

[0016] According to another aspect, an image decoder is provided including: an entropy decoder which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients; an inverse quantizer and transformer which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals; an adder which, in operation, adds the residuals outputted from the inverse quantizer and transformer and predictions outputted from a prediction controller to reconstruct blocks; and the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an decoded reference block in a current picture. The prediction controller, in operation, splits an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicts a first motion vector for the first partition and a second motion vector for the second partition; and decodes the first partition using the first motion vector and the second partition using the second motion vector.

[0017] According to another aspect, an image decoding method is provided, which includes generally three steps: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicting a first motion vector for the first partition and a second motion vector for the second partition; and decoding the first partition using the first motion vector and the second partition using the second motion vector.

[0018] According to one aspect, an image encoder is provided including circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition; second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition; weighting the first values and the second values; and encoding the first partition using the weighted first values and the weighted second values.

[0019] According to a further aspect, the non-rectangular shape is a triangle. According to another aspect, the non-rectangular shape is selected from a group consisting of a triangle, a trapezoid, and a polygon with at least five sides and angles. According to yet another aspect, the second partition has a non-rectangular shape.

[0020] According to another aspect, at least one of the first-predicting and the second-predicting is an inter prediction process that predicts the first values and the second values based on a reference partition in an encoded reference picture. The inter-prediction process may predict first values of pixels of the first partition including the set of pixels and may predict the second values of only the set of pixels of the first partition.

[0021] According to another aspect, at least one of the first-predicting and the second-predicting is an intra prediction process that predicts the first values and the second values based on an encoded reference partition in a current picture.

[0022] According to another aspect, a prediction method used in the first-predicting is different from a prediction method used in the second-predicting.

[0023] According to a further aspect, a number of the set of pixels of each row or each column, for which the first values and the second values are predicted, is an integer. For example, when the number of the set of pixels of each row or each column is four, weights of 1/8, 1/4, 3/4, and 7/8 may be applied to the first values of the four pixels in the set, respectively, and weights of 7/8, 3/4, 1/4, and 1/8 may be applied to the second values of the four pixels in the set, respectively. As another example, when the number of the set of pixels of each row or each column is two, weights of 1/3 and 2/3 may be applied to the first values of the two pixels in the set, respectively, and weights of 2/3 and 1/3 may be applied to the second values of the two pixels in the set, respectively.

[0024] According to another aspect, the weights may be integer values or may be fractional values.

[0025] According to another aspect, an image encoder is provided including: a splitter which, in operation, receives and splits an original picture into blocks; an adder which, in operation, receives the blocks from the splitter and predictions from a prediction controller, and subtracts each prediction from its corresponding block to output a residual; a transformer which, in operation, performs a transform on the residuals outputted from the adder to output transform coefficients; a quantizer which, in operation, quantizes the transform coefficients to generate quantized transform coefficients; an entropy encoder which, in operation, encodes the quantized transform coefficients to generate a bitstream; and the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an encoded reference block in a current picture. The prediction controller, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition; second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition; weighting the first values and the second values; and encoding the first partition using the weighted first values and the weighted second values.

[0026] According to another aspect, an image encoding method is provided to perform a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. The method includes generally four steps: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition; second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition; weighting the first values and the second values; and encoding the first partition using the weighted first values and the weighted second values.

[0027] According to a further aspect, an image decoder is provided which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition; second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition; weighting the first values and the second values; and decoding the first partition using the weighted first values and the weighted second values.

[0028] According to another aspect, the non-rectangular shape is a triangle. According to a further aspect, the non-rectangular shape is selected from a group consisting of a triangle, a trapezoid, and a polygon with at least five sides and angles. According to another aspect, the second partition has a non-rectangular shape.

[0029] According to another aspect, at least one of the first-predicting and the second-predicting is an inter prediction process that predicts the first values and the second values based on a reference partition in an encoded reference picture. The inter-prediction process may predict first values of pixels of the first partition including the set of pixels and may predict the second values of only the set of pixels of the first partition.

[0030] According to another aspect, at least one of the first-predicting and the second-predicting is an intra prediction process that predicts the first values and the second values based on an encoded reference partition in a current picture.

[0031] According to another aspect, an image decoder is provided including: an entropy decoder which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients; an inverse quantizer and transformer which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals; an adder which, in operation, adds the residuals outputted from the inverse quantizer and transformer and predictions outputted from a prediction controller to reconstruct blocks; and the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an decoded reference block in a current picture. The prediction controller, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition; second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition; weighting the first values and the second values; and decoding the first partition using the weighted first values and the weighted second values.

[0032] According to another aspect, an image decoding method is provided to perform a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. The method includes generally four steps: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition; second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition; weighting the first values and the second values; and decoding the first partition using the weighted first values and the weighted second values.

[0033] According to one aspect, an image encoder is provided including circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a partition syntax operation including: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition based on a partition parameter indicative of the splitting; encoding the first partition and the second partition; and writing one or more parameters including the partition parameter into a bitstream.

[0034] According to a further aspect, the partition parameter indicates the first partition has a triangle shape.

[0035] According to another aspect, the partition parameter indicates the second partition has a non-rectangular shape.

[0036] According to another aspect, the partition parameter indicates the non-rectangular shape is one of a triangle, a trapezoid, and a polygon with at least five sides and angles.

[0037] According to another aspect, the partition parameter jointly encodes a split direction applied to split the image block into the plurality of partitions. For example, the split direction may include: from a top-left corner of the image block to a bottom-right corner thereof, and from a top-right corner of the image block to a bottom-left corner thereof. The partition parameter may jointly encode at least a first motion vector of the first partition.

[0038] According to another aspect, the one or more parameters other than the partition parameter encodes a split direction applied to split the image block into the plurality of partitions. The parameter encoding the split direction may jointly encode at least a first motion vector of the first partition.

[0039] According to another aspect, the partition parameter may jointly encode at least a first motion vector of the first partition. The partition parameter may jointly encode a second motion vector of the second partition.

[0040] According to another aspect, the one or more parameters other than the partition parameter may encode at least a first motion vector of the first partition.

[0041] According to another aspect, the one or more parameters are binarized pursuant to a binarization scheme which is selected depending on a value of at least one of the one or more parameters.

[0042] According to a further aspect, an image encoder is provided including: a splitter which, in operation, receives and splits an original picture into blocks; an adder which, in operation, receives the blocks from the splitter and predictions from a prediction controller, and subtracts each prediction from its corresponding block to output a residual; a transformer which, in operation, performs a transform on the residuals outputted from the adder to output transform coefficients; a quantizer which, in operation, quantizes the transform coefficients to generate quantized transform coefficients; an entropy encoder which, in operation, encodes the quantized transform coefficients to generate a bitstream; and the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an encoded reference block in a current picture. The prediction controller, in operation, splits an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition based on a partition parameter indicative of the splitting, and encodes the first partition and the second partition. The entropy encoder, in operation, writes one or more parameters including the partition parameter into a bitstream.

[0043] According to another aspect, an image encoding method including a partition syntax operation is provided. The method includes generally three steps: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition based on a partition parameter indicative of the splitting; encoding the first partition and the second partition; and writing one or more parameters including the partition parameter into a bitstream.

[0044] According to another aspect, an image decoder is provided including circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a partition syntax operation including: parsing one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; splitting the image block into the plurality of partitions based on the partition parameter; and decoding the first partition and the second partition.

[0045] According to a further aspect, the partition parameter indicates the first partition has a triangle shape.

[0046] According to another aspect, the partition parameter indicates the second partition has a non-rectangular shape.

[0047] According to another aspect, the partition parameter indicates the non-rectangular shape is one of a triangle, a trapezoid, and a polygon with at least five sides and angles.

[0048] According to another aspect, the partition parameter jointly encodes a split direction applied to split the image block into the plurality of partitions. For example, the split direction includes: from a top-left corner of the image block to a bottom-right corner thereof, and from a top-right corner of the image block to a bottom-left corner thereof. The partition parameter may jointly encode at least a first motion vector of the first partition.

[0049] According to another aspect, the one or more parameters other than the partition parameter encodes a split direction applied to split the image block into the plurality of partitions. The parameter encoding the split direction may jointly encode at least a first motion vector of the first partition.

[0050] According to another aspect, the partition parameter may jointly encode at least a first motion vector of the first partition. The partition parameter may jointly encode a second motion vector of the second partition.

[0051] According to another aspect, the one or more parameters other than the partition parameter may encode at least a first motion vector of the first partition.

[0052] According to another aspect, the one or more parameters are binarized pursuant to a binarization scheme which is selected depending on a value of at least one of the one or more parameters.

[0053] According to a further aspect, an image decoder is provided including: an entropy decoder which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients; an inverse quantizer and transformer which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals; an adder which, in operation, adds the residuals outputted from the inverse quantizer and transformer and predictions outputted from a prediction controller to reconstruct blocks; and the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an decoded reference block in a current picture. The entropy decoder, in operation: parses one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; splits the image block into the plurality of partitions based on the partition parameter; and decodes the first partition and the second partition.

[0054] According to another aspect, an image decoding method including a partition syntax operation is provided. The method includes generally three steps: parsing one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; splitting the image block into the plurality of partitions based on the partition parameter; and decoding the first partition and the second partition.

[0055] In the drawings, identical reference numbers identify similar elements. The sizes and relative positions of elements in the drawings are not necessarily drawn to scale.

[0056] Hereinafter, embodiment(s) will be described with reference to the drawings. Note that the embodiment(s) described below each show a general or specific example. The numerical values, shapes, materials, components, the arrangement and connection of the components, steps, the relation and order of the steps, etc., indicated in the following embodiment(s) are mere examples, and are not intended to limit the scope of the claims. Therefore, those components disclosed in the following embodiment(s) but not recited in any of the independent claims defining the broadest inventive concepts may be understood as optional components.

[0057] Embodiments of an encoder and a decoder will be described below. The embodiments are examples of an encoder and a decoder to which the processes and/or configurations presented in the description of aspects of the present disclosure are applicable. The processes and/or configurations can also be implemented in an encoder and a decoder different from those according to the embodiments. For example, regarding the processes and/or configurations as applied to the embodiments, any of the following may be implemented:

[0058] (1) Any of the components of the encoder or the decoder according to the embodiments presented in the description of aspects of the present disclosure may be substituted or combined with another component presented anywhere in the description of aspects of the present disclosure.

[0059] (2) In the encoder or the decoder according to the embodiments, discretionary changes may be made to functions or processes performed by one or more components of the encoder or the decoder, such as addition, substitution, removal, etc., of the functions or processes. For example, any function or process may be substituted or combined with another function or process presented anywhere in the description of aspects of the present disclosure.

[0060] (3) In the method implemented by the encoder or the decoder according to the embodiments, discretionary changes may be made such as addition, substitution, and removal of one or more of the processes included in the method. For example, any process in the method may be substituted or combined with another process presented anywhere in the description of aspects of the present disclosure.

[0061] (4) One or more components included in the encoder or the decoder according to embodiments may be combined with a component presented anywhere in the description of aspects of the present disclosure, may be combined with a component including one or more functions presented anywhere in the description of aspects of the present disclosure, and may be combined with a component that implements one or more processes implemented by a component presented in the description of aspects of the present disclosure.

[0062] (5) A component including one or more functions of the encoder or the decoder according to the embodiments, or a component that implements one or more processes of the encoder or the decoder according to the embodiments, may be combined or substituted with a component presented anywhere in the description of aspects of the present disclosure, with a component including one or more functions presented anywhere in the description of aspects of the present disclosure, or with a component that implements one or more processes presented anywhere in the description of aspects of the present disclosure.

[0063] (6) In the method implemented by the encoder or the decoder according to the embodiments, any of the processes included in the method may be substituted or combined with a process presented anywhere in the description of aspects of the present disclosure or with any corresponding or equivalent process.

[0064] (7) One or more processes included in the method implemented by the encoder or the decoder according to the embodiments may be combined with a process presented anywhere in the description of aspects of the present disclosure.

[0065] (8) The implementation of the processes and/or configurations presented in the description of aspects of the present disclosure is not limited to the encoder or the decoder according to the embodiments. For example, the processes and/or configurations may be implemented in a device used for a purpose different from the moving picture encoder or the moving picture decoder disclosed in the embodiments.

[0066] (Encoder)

First, the encoder according to an embodiment will be described. FIG. 1 is a block diagram illustrating a functional configuration of encoder 100 according to the embodiment. Encoder 100 is a moving picture encoder that encodes a moving picture block by block.

[0067] As illustrated in FIG. 1, encoder 100 is a device that encodes a picture block by block, and includes splitter 102, subtractor 104, transformer 106, quantizer 108, entropy encoder 110, inverse quantizer 112, inverse transformer 114, adder 116, block memory 118, loop filter 120, frame memory 122, intra predictor 124, inter predictor 126, and prediction controller 128.

[0068] Encoder 100 is realized as, for example, a generic processor and memory. In this case, when a software program stored in the memory is executed by the processor, the processor functions as splitter 102, subtractor 104, transformer 106, quantizer 108, entropy encoder 110, inverse quantizer 112, inverse transformer 114, adder 116, loop filter 120, intra predictor 124, inter predictor 126, and prediction controller 128. Alternatively, encoder 100 may be realized as one or more dedicated electronic circuits corresponding to splitter 102, subtractor 104, transformer 106, quantizer 108, entropy encoder 110, inverse quantizer 112, inverse transformer 114, adder 116, loop filter 120, intra predictor 124, inter predictor 126, and prediction controller 128.

[0069] Hereinafter, each component included in encoder 100 will be described.

[0070] (Splitter)

Splitter 102 splits each picture included in an inputted moving picture into blocks, and outputs each block to subtractor 104. For example, splitter 102 first splits a picture into blocks of a fixed size (for example, 128×128). The fixed size block may also be referred to as a coding tree unit (CTU). Splitter 102 then splits each fixed size block into blocks of variable sizes (for example, 64×64 or smaller) based, for example, on recursive quadtree and/or binary tree block splitting. The variable size block may also be referred to as a coding unit (CU), a prediction unit (PU), or a transform unit (TU). In various implementations there may be no need to differentiate between CU, PU, and TU; all or some of the blocks in a picture may be processed per CU, PU, or TU.

[0071] FIG. 2 illustrates one example of block splitting according to an embodiment. In FIG. 2, the solid lines represent block boundaries of blocks split by quadtree block splitting, and the dashed lines represent block boundaries of blocks split by binary tree block splitting.

[0072] Here, block 10 is a square 128×128 pixel block (128×128 block). This 128×128 block 10 is first split into four square 64×64 blocks (quadtree block splitting).

[0073] The top left 64×64 block is further vertically split into two rectangle 32×64 blocks, and the left 32×64 block is further vertically split into two rectangle 16×64 blocks (binary tree block splitting). As a result, the top left 64×64 block is split into two 16×64 blocks 11 and 12 and one 32×64 block 13.

[0074] The top right 64×64 block is horizontally split into two rectangle 64×32 blocks 14 and 15 (binary tree block splitting).

[0075] The bottom left 64×64 block is first split into four square 32×32 blocks (quadtree block splitting). The top left block and the bottom right block among the four 32×32 blocks are further split. The top left 32×32 block is vertically split into two rectangle

16×32 blocks, and the right 16×32 block is further horizontally split into two 16×16 blocks (binary tree block splitting). The bottom right 32×32 block is horizontally split into two 32×16 blocks (binary tree block splitting). As a result, the bottom left 64×64 block is split into 16×32 block 16, two 16×16 blocks 17 and 18, two 32×32 blocks 19 and 20, and two 32×16 blocks 21 and 22.

[0076] The bottom right 64×64 block 23 is not split.

[0077] As described above, in FIG. 2, block 10 is split into 13 variable size blocks 11 through 23 based on recursive quadtree and binary tree block splitting. This type of splitting is also referred to as quadtree plus binary tree (QTBT) splitting.

[0078] While in FIG. 2 one block is split into four or two blocks (quadtree or binary tree block splitting), splitting is not limited to these examples. For example, one block may be split into three blocks (ternary block splitting). Splitting including such ternary block splitting is also referred to as multi-type tree (MBT) splitting.

[0079] (Subtractor)

Subtractor 104 subtracts a prediction signal (prediction sample, inputted from prediction controller 128, to be described below) from an original signal (original sample) per block split by and inputted from splitter 102. In other words, subtractor 104 calculates prediction errors (also referred to as "residuals") of a block to be encoded (hereinafter referred to as a "current block"). Subtractor 104 then outputs the calculated prediction errors (residuals) to transformer 106.

[0080] The original signal is a signal input into encoder 100, and is a signal representing an image for each picture included in a moving picture (for example, a luma signal and two chroma signals). Hereinafter, a signal representing an image is also referred to as a sample.

[0081] (Transformer)

Transformer 106 transforms spatial domain prediction errors into frequency domain transform coefficients, and outputs the transform coefficients to quantizer 108. More specifically, transformer 106 applies, for example, a predefined discrete cosine transform (DCT) or discrete sine transform (DST) to spatial domain prediction errors.

[0082] Note that transformer 106 may adaptively select a transform type from among a plurality of transform types, and transform prediction errors into transform coefficients by using a transform basis function corresponding to the selected transform type. This sort of transform is also referred to as explicit multiple core transform (EMT) or adaptive multiple transform (AMT).

[0083] The transform types include, for example, DCT-II, DCT-V, DCT-VIII, DST-I, and DST-VII. FIG. 3 is a chart indicating transform basis functions for each transform type. In FIG. 3, N indicates the number of input pixels. For example, selection of a transform type from among the plurality of transform types may depend on the prediction type (intra prediction and inter prediction) as well as intra prediction mode.

[0084] Information indicating whether to apply EMT or AMT (referred to as, for example, an EMT flag or an AMT flag) and information indicating the selected transform type is typically signaled at the CU level. Note that the signaling of such information need not be performed at the CU level, and may be performed at another level (for example, at the bit sequence level, picture level, slice level, tile level, or CTU level).

[0085] Moreover, transformer 106 may apply a secondary transform to the transform coefficients (transform result). Such a secondary transform is also referred to as adaptive secondary transform (AST) or non-separable secondary transform (NSST). For example, transformer 106 applies a secondary transform to each sub-block (for example, each 4×4 sub-block) included in the block of the transform coefficients corresponding to the intra prediction errors. Information indicating whether to apply NSST and information related to the transform matrix used in NSST are typically signaled at the CU level. Note that the signaling of such information need not be performed at the CU level, and may be performed at another level (for example, at the sequence level, picture level, slice level, tile level, or CTU level).

[0086] Either a separate transform or a non-separable transform may be applied in transformer 106. A separate transform is a method in which a transform is performed a plurality of times by separately performing a transform for each direction according to the number of dimensions input. A non-separable transform is a method of performing a collective transform in which two or more dimensions in a multidimensional input are collectively regarded as a single dimension.

[0087] In one example of a non-separable transform, when the input is a 4×4 block, the 4×4 block is regarded as a single array including 16 components, and the transform applies a 16×16 transform matrix to the array.

[0088] In a further example of a non-separable transform, after the input 4×4 block is regarded as a single array including 16 components, a transform that performs a plurality of Givens rotations (e.g., a Hypercube-Givens Transform) may be applied on the array.

[0089] (Quantizer)

Quantizer 108 quantizes the transform coefficients output from transformer 106. More specifically, quantizer 108 scans, in a predetermined scanning order, the transform coefficients of the current block, and quantizes the scanned transform coefficients based on quantization parameters (QP) corresponding to the transform coefficients. Quantizer 108 then outputs the quantized transform coefficients (hereinafter referred to as quantized coefficients) of the current block to entropy encoder 110 and inverse quantizer 112.

[0090] A predetermined scanning order is an order for quantizing/inverse quantizing transform coefficients. For example, a predetermined scanning order is defined as ascending order of frequency (from low to high frequency) or descending order of frequency (from high to low frequency).

[0091] A quantization parameter (QP) is a parameter defining a quantization step size (quantization width). For example, if the value of the quantization parameter increases, the quantization step size also increases. In other words, if the value of the quantization parameter increases, the quantization error increases.

[0092] (Entropy Encoder)

Entropy encoder 110 generates an encoded signal (encoded bitstream) based on the quantized coefficients, which are inputted from quantizer 108. More specifically, for example, entropy encoder 110 binarizes quantized coefficients and arithmetic encodes the binary signal, to output a compressed bitstream or sequence.

[0093] (Inverse Quantizer)

Inverse quantizer 112 inverse quantizes the quantized coefficients, which are inputted from quantizer 108. More specifically, inverse quantizer 112 inverse quantizes, in a predetermined scanning order, quantized coefficients of the current block. Inverse quantizer 112 then outputs the inverse quantized transform coefficients of the current block to inverse transformer 114.

[0094] (Inverse Transformer)

Inverse transformer 114 restores prediction errors (residuals) by inverse transforming the transform coefficients, which are inputted from inverse quantizer 112. More specifically, inverse transformer 114 restores the prediction errors of the current block by applying an inverse transform corresponding to the transform applied by transformer 106 on the transform coefficients. Inverse transformer 114 then outputs the restored prediction errors to adder 116.

[0095] Note that since, typically, information is lost in quantization, the restored prediction errors do not match the prediction errors calculated by subtractor 104. In other words, the restored prediction errors typically include quantization errors.

[0096] (Adder)

Adder 116 reconstructs the current block by summing prediction errors, which are inputted from inverse transformer 114, and prediction samples, which are inputted from prediction controller 128. Adder 116 then outputs the reconstructed block to block memory 118 and loop filter 120. A reconstructed block is also referred to as a local decoded block.

[0097] (Block Memory)

Block memory 118 is storage for storing blocks in a picture to be encoded (referred to as a "current picture") for reference in intra prediction, for example. More specifically, block memory 118 stores reconstructed blocks output from adder 116.

[0098] (Loop Filter)

Loop filter 120 applies a loop filter to blocks reconstructed by adder 116, and outputs the filtered reconstructed blocks to frame memory 122. A loop filter is a filter used in an encoding loop (in-loop filter), and includes, for example, a deblocking filter (DF), a sample adaptive offset (SAO), and an adaptive loop filter (ALF).

[0099] In ALF, a least square error filter for removing compression artifacts is applied. For example, one filter from among a plurality of filters is selected for each 2×2 sub-block in the current block based on direction and activity of local gradients, and is applied.

[0100] More specifically, first, each sub-block (for example, each 2×2 sub-block) is categorized into one out of a plurality of classes (for example, 15 or 25 classes). The classification of the sub-block is based on gradient directionality and activity. For example, classification index C is derived based on gradient directionality D (for example, 0 to 2 or 0 to 4) and gradient activity A (for example, 0 to 4) (for example, $C = 5D + A$). Then, based on classification index C, each sub-block is categorized into one out of a plurality of classes.

[0101] For example, gradient directionality D is calculated by comparing gradients of a plurality of directions (for example, the horizontal, vertical, and two diagonal directions). Furthermore, for example, gradient activity A is calculated by summing gradients of a plurality of directions and quantizing the sum.

[0102] The filter to be used for each sub-block is determined from among the plurality of filters based on the result of such categorization.

[0103] The filter shape to be used in ALF is, for example, a circular symmetric filter shape. FIGS. 4A, 4B, and 4C illustrate examples of filter shapes used in ALF. FIG. 4A illustrates a 5×5 diamond shape filter, FIG. 4B illustrates a 7×7 diamond shape filter, and FIG. 4C illustrates a 9×9 diamond shape filter. Information indicating the filter shape is typically signaled at the picture level. Note that the signaling of information indicating the filter shape need not be performed at the picture level, and may be performed at another level (for example, at the sequence level, slice level, tile level, CTU level, or CU level).

[0104] The enabling or disabling of ALF may be determined at the picture level or CU level. For example, for luma, the decision to apply ALF or not may be done at the CU level, and for chroma, the decision to apply ALF or not may be done at the picture level. Information indicating whether ALF is enabled or disabled is typically signaled at the picture level or CU level. Note that the signaling of information indicating whether ALF is enabled or disabled need not be performed at the picture level or CU level, and may be performed at another level (for example, at the sequence level, slice level, tile level, or CTU level).

[0105] The coefficients set for the plurality of selectable filters (for example, 15 or 25 filters) is typically signaled at the picture level. Note that the signaling of the coefficients set need not be performed at the picture level, and may be performed at another level (for example, at the sequence level, slice level, tile level, CTU level, CU level, or sub-block level).

[0106] (Frame Memory)

Frame memory 122 is storage for storing reference pictures used in inter prediction, for example, and is also referred to as a frame buffer. More specifically, frame memory 122 stores reconstructed blocks filtered by loop filter 120.

[0107] (Intra Predictor)

Intra predictor 124 generates a prediction signal (intra prediction signal) by intra predicting the current block with reference to a block or blocks that are in the current picture as stored in block memory 118 (also referred to as intra frame prediction). More specifically, intra predictor 124 generates an intra prediction signal by intra prediction with reference to samples (for example, luma and/or chroma values) of a block or blocks neighboring the current block, and then outputs the intra prediction signal to prediction controller 128.

[0108] For example, intra predictor 124 performs intra prediction by using one mode from among a plurality of predefined intra prediction modes. The intra prediction modes typically include one or more non-directional prediction modes and a plurality of directional prediction modes.

[0109] The one or more non-directional prediction modes include, for example, planar prediction mode and DC prediction mode defined in the H.265/HEVC standard.

[0110] The plurality of directional prediction modes include, for example, the 33 directional prediction modes defined in the H.265/HEVC standard. Note that the plurality of directional prediction modes may further include 32 directional prediction modes in addition to the 33 directional prediction modes (for a total of 65 directional prediction modes).

[0111] FIG. 5A illustrates a total of 67 intra prediction modes used in intra prediction (two non-directional prediction modes and 65 directional prediction modes). The solid arrows represent the 33 directions defined in the H.265/HEVC standard, and the dashed arrows represent the additional 32 directions. (The two "non-directional" prediction modes are not illustrated in FIG. 5A.)

[0112] In various implementations, a luma block may be referenced in chroma block intra prediction. That is, a chroma component of the current block may be predicted based on a luma component of the current block. Such intra prediction is also referred to as cross-component linear model (CCLM) prediction. The chroma block intra prediction mode that references a luma block (referred to as, for example, CCLM mode) may be added as one of the chroma block intra prediction modes.

[0113] Intra predictor 124 may correct post-intra-prediction pixel values based on horizontal/vertical reference pixel gradients. Intra prediction accompanied by this sort of correcting is also referred to as position dependent intra prediction combination (PDPC). Information indicating whether to apply PDPC or not (referred to as, for example, a PDPC flag) is typically signaled at the CU level. Note that the signaling of this information need not be performed at the CU level, and may be performed at another level (for example, on the sequence level, picture level, slice level, tile level, or CTU level).

[0114] (Inter Predictor)

Inter predictor 126 generates a prediction signal (inter prediction signal) by inter predicting the current block with reference to a block or blocks in a reference picture, which is different from the current picture and is stored in frame memory 122 (also referred to as inter frame prediction). Inter prediction is performed per current block or per current sub-block (for example, per 4×4 block) in the current block. For example, inter predictor 126 performs motion estimation in a reference picture for the current block or the current sub-block, to find a reference block or sub-block in the reference picture that best matches the current block or sub-block, and to obtain motion information (for example, a motion vector) that compensates for (or predicts) the movement or change from the reference block or sub-block to the current block or sub-block. Inter predictor 126 then performs motion compensation (or motion prediction) based on the motion information, and generates an inter prediction signal of the current block or sub-block based on the motion information. Inter predictor 126 then outputs the generated inter prediction signal to prediction controller 128.

[0115] The motion information used in motion compensation may be signaled in a variety of forms as the inter prediction signal. For example, a motion vector may be signaled. As another example, a difference between a motion vector and a motion vector predictor may be signaled.

[0116] Note that the inter prediction signal may be generated using motion information for a neighboring block in addition to motion information for the current block obtained from motion estimation. More specifically, the inter prediction signal may be generated per sub-block in the current block by calculating a weighted sum of a prediction signal based on motion information obtained from the motion estimation (in the reference picture) and a prediction signal based on motion information of a neighboring block (in the current picture). Such inter prediction (motion compensation) is also referred to as overlapped block motion compensation (OBMC).

[0117] In OBMC mode, information indicating sub-block size for OBMC (referred to as, for example, OBMC block size) may be signaled at the sequence level. Further, information indicating whether to apply the OBMC mode or not (referred to as, for example, an OBMC flag) may be signaled at the CU level. Note that the signaling of such information need not be performed at the sequence level and CU level, and may be performed at another level (for example, at the picture level, slice level, tile level, CTU level, or sub-block level).

[0118] Hereinafter, the OBMC mode will be described in further detail. FIG. 5B is a flowchart and FIG. 5C is a conceptual diagram illustrating a prediction image correction process performed by OBMC processing.

[0119] Referring to FIG. 5C, first, a prediction image (Pred) is obtained through typical motion compensation using a motion vector (MV) assigned to the target (current) block. In FIG. 5C, an arrow "MV" points to the reference picture, to indicate what the current block in the current picture is referencing in order to obtain a prediction image.

[0120] Next, a prediction image (Pred_L) is obtained by applying (reusing) a motion vector (MV_L), which was already derived for the encoded neighboring left block, to the target (current) block, as indicated by an arrow "MV_L" originating from the current block and pointing to the reference picture to obtain the prediction image Pred_L. Then, the two prediction images Pred and Pred_L are superimposed to perform a first pass of the correction of the prediction image, which in one aspect has an effect of blending the border between the neighboring blocks.

[0121] Similarly, a prediction image (Pred_U) is obtained by applying (reusing) a motion vector (MV_U), which was already derived for the encoded neighboring upper block, to the target (current) block, as indicated by an arrow "MV_U" originating from the current block and pointing to the reference picture to obtain the prediction image Pred_U. Then, the prediction image Pred_U is superimposed with the prediction image resulting from the first pass (i.e., Pred and Pred_L) to perform a second pass of the correction of the prediction image, which in one aspect has an effect of blending the border between the neighboring blocks. The result of the second pass is the final prediction image for the current block, with blended (smoothed) borders with its neighboring blocks.

[0122] Note that the above example is of a two-pass correction method using the neighboring left and upper blocks, but the method may be a three-pass or higher-pass correction method that also uses the neighboring right and/or lower block.

[0123] Note that the region subject to superimposition may be the entire pixel region of the block, and, alternatively, may be a partial block boundary region.

[0124] Note that here, the prediction image correction process of OBMC is described as being based on a single reference picture to derive a single prediction image Pred, to which additional prediction images Pred_L and Pred_U are superimposed, but the same process may apply to each of a plurality of reference pictures when the prediction image is corrected based on the plurality of reference pictures. In such a case, after a plurality of corrected prediction images are obtained by performing the image correction of OBMC based on the plurality of reference pictures, respectively, the obtained plurality of corrected prediction images are further superimposed to obtain the final prediction image.

[0125] Note that, in OBMC, the unit of the target block may be a prediction block and, alternatively, may be a sub-block obtained by further dividing the prediction block.

[0126] One example of a method to determine whether to implement OBMC processing is to use an obmc_flag, which is a signal that indicates whether to implement OBMC processing. As one specific example, the encoder may determine whether the target block belongs to a region including complicated motion. The encoder sets the obmc_flag to a value of "1" when the block belongs to a region including complicated motion and implements OBMC processing during encoding, and sets the obmc_flag to a value of "0" when the block does not belong to a region including complication motion and encodes the block without implementing OBMC processing. The decoder switches between implementing OBMC processing or not by decoding the obmc_flag written in the stream (i.e., the compressed sequence) and performing the decoding in accordance with the flag value.

[0127] Note that the motion information may be derived on the decoder side without being signaled from the encoder side. For example, a merge mode defined in the H.265/HEVC standard may be used. Furthermore, for example, the motion information may be derived by performing motion estimation on the decoder side. In this case, the decoder side may perform motion estimation without using the pixel values of the current block.

[0128] Here, a mode for performing motion estimation on the decoder side will be described. A mode for performing motion estimation on the decoder side is also referred to as pattern matched motion vector derivation (PMMVD) mode or frame rate up-conversion (FRUC) mode.

[0129] One example of FRUC processing is illustrated in FIG. 5D. First, a candidate list (a candidate list may be a merge list) of candidates, each including a prediction motion vector (MV), is generated with reference to motion vectors of encoded blocks that spatially or temporally neighbor the current block. Next, the best candidate MV is selected from among the plurality of candidate MVs registered in the candidate list. For example, evaluation values for the candidate MVs included in the candidate list are calculated and one candidate MV is selected based on the calculated evaluation values.

[0130] Next, a motion vector for the current block is derived from the motion vector of the selected candidate. More specifically, for example, the motion vector for the current block is calculated as the motion vector of the selected candidate (the best candidate MV), as-is. Alternatively, the motion vector for the current block may be derived by pattern matching performed in the vicinity of a position in a reference picture corresponding to the motion vector of the selected candidate. In other words, when the vicinity of the best candidate MV is searched using pattern matching in a reference picture and evaluation values, and an MV having a better evaluation value is found, the best candidate MV may be updated to the MV having the better evaluation value, and the MV having the better evaluation value may be used as the final MV for the current block. A configuration in which the processing to update the MV having a better evaluation value is not implemented is also acceptable.

[0131] The same processes may be performed in cases in which the processing is performed in units of sub-blocks.

[0132] An evaluation value may be calculated in various ways. For example, a reconstructed image of a region in a reference picture corresponding to a motion vector is compared with a reconstructed image of a predetermined region (which may be in another reference picture or in a neighboring block in the current picture, for example, as described below), and a difference in pixel values between the two reconstructed images may be calculated and used as an evaluation value of the motion vector. Note that the evaluation value may be calculated by using some other information in addition to the difference.

[0133] Next, pattern matching is described in detail. First, one candidate MV included in a candidate list (e.g., a merge list) is selected as the starting point for the search by pattern matching. The pattern matching used is either first pattern matching or second pattern matching. First pattern matching and second pattern matching are also referred to as bilateral matching and template matching, respectively.

[0134] In first pattern matching, pattern matching is performed between two blocks in two different reference pictures that are both along the motion trajectory of the current block. Therefore, in first pattern matching, for a region in a reference picture, a region in another reference picture that conforms to the motion trajectory of the current block is used as the predetermined region for the above-described calculation of the candidate's evaluation value.

[0135] FIG. 6 illustrates one example of first pattern matching (bilateral matching) between two blocks in two reference pictures along a motion trajectory. As illustrated in FIG. 6, in first pattern matching, two motion vectors (MV0, MV1) are derived by finding the best match between the two blocks in two different reference pictures (Ref0, Ref1) along the motion trajectory of the current block (Cur block). More specifically, a difference may be obtained between (i) a reconstructed image at a position specified by a candidate MV in a first encoded reference picture (Ref0), and (ii) a reconstructed image at a position specified by the candidate MV, which is symmetrically scaled per display time intervals, in a second encoded reference picture (Ref1). Then, the difference may be used to derive an evaluation value for the current block. A candidate MV having the best evaluation value among a plurality of candidate MVs may be selected as the final MV.

[0136] Under the assumption of continuous motion trajectory, the motion vectors (MV0, MV1) pointing to the two reference blocks are proportional to the temporal distances (TD0, TD1) between the current picture (Cur Pic) and the two reference pictures (Ref0, Ref1). For example, when the current picture is temporally between the two reference pictures, and the temporal distance from the current picture to the two reference pictures is the same, first pattern matching derives two mirroring bi-directional motion vectors.

[0137] In second pattern matching (template matching), pattern matching is performed between a template in the current picture (blocks neighboring the current block in the current picture; for example, the top and/or left neighboring blocks) and a block in a reference picture. Therefore, in second pattern matching, a block neighboring the current block in the current picture is used as the predetermined region for the above-described calculation of the candidate evaluation value.

[0138] FIG. 7 illustrates one example of pattern matching (template matching) between a template in the current picture and a block in a reference picture. As illustrated in FIG. 7, in second pattern matching, a motion vector of the current block is derived by searching in a reference picture (Ref0) to find a block that best matches neighboring block(s) of the current block (Cur block) in the current picture (Cur Pic). More specifically, a difference may be obtained between (i) a reconstructed image of one or both of encoded neighboring upper and left regions relative to the current block, and (ii) a reconstructed image of the same regions relative to a block position specified by a candidate MV in an encoded reference picture (Ref0). Then, the difference may be used to derive an evaluation value for the current block. A candidate MV having the best evaluation value among a plurality of candidate MVs may be selected as the best candidate MV.

[0139] Information indicating whether to apply the FRUC mode or not (referred to as, for example, a FRUC flag) may be signaled at the CU level. Further, when the FRUC mode is applied (for example, when the FRUC flag is set to true), information indicating the pattern applicable matching method (e.g., first pattern matching or second pattern matching) may be signaled at the CU level. Note that the signaling of such information need not be performed at the CU level, and may be performed at another level (for example, at the sequence level, picture level, slice level, tile level, CTU level, or sub-block level).

[0140] Next, methods of deriving a motion vector are described. First, a description is given of a mode for deriving a motion vector based on a model assuming uniform linear motion. This mode is also referred to as a bi-directional optical flow (BIO) mode.

[0141] FIG. 8 illustrates a model that assumes uniform linear motion. In FIG. 8, (v_x, v_y) denotes a velocity vector, and τ_0 and τ_1 denote temporal distances between the current picture (Cur Pic) and two reference pictures (Ref_0, Ref_1), respectively. (MVx_0, MVy_0) denotes a motion vector corresponding to reference picture Ref_0, and (MVx_1, MVy_1) denotes a motion vector corresponding to reference picture Ref_1.

[0142] Here, under the assumption of uniform linear motion exhibited by velocity vector (v_x, v_y), (MVx_0, MVy_0) and (MVx_1, MVy_1) are represented as $(v_x\tau_0, v_y\tau_0)$ and $(-v_x\tau_1, -v_y\tau_1)$, respectively, and the following optical flow equation (Equation 1) is given.

[0143] [Math.1]

$$\partial I^{(k)}/\partial t + v_x\, \partial I^{(k)}/\partial x + v_y\, \partial I^{(k)}/\partial y = 0. \qquad (1)$$

[0144] Here, $I^{(k)}$ denotes a luma value from reference picture k (k = 0, 1) after motion compensation. The optical flow equation shows that the sum of (i) the time derivative of the luma value, (ii) the product of the horizontal velocity and the horizontal component of the spatial gradient of a reference picture, and (iii) the product of the vertical velocity and the vertical component of the spatial gradient of a reference picture, is equal to zero. A motion vector of each block obtained from, for example, a merge list may be corrected pixel by pixel based on a combination of the optical flow equation and Hermite interpolation.

[0145] Note that a motion vector may be derived on the decoder side using a method other than deriving a motion vector based on a model assuming uniform linear motion. For example, a motion vector may be derived for each sub-block based on motion vectors of neighboring blocks.

[0146] Next, a description is given of a mode in which a motion vector is derived for each sub-block based on motion vectors of neighboring blocks. This mode is also referred to as affine motion compensation prediction mode.

[0147] FIG. 9A illustrates one example of deriving a motion vector of each sub-block based on motion vectors of neighboring blocks. In FIG. 9A, the current block includes 16 4×4 sub-blocks. Here, motion vector v_0 of the top left corner control point in the current block is derived based on motion vectors of neighboring sub-blocks. Similarly, motion vector v_1 of the top right corner control point in the current block is derived based on motion vectors of neighboring blocks. Then, using the two motion vectors v_0 and v_1, the motion vector (v_x, v_y) of each sub-block in the current block is derived using Equation 2 below.

[0148]

[Math.2]

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w}x - \frac{(v_{1y} - v_{0y})}{w}y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w}x + \frac{(v_{1x} - v_{0x})}{w}y + v_{0y} \end{cases} \quad (2)$$

[0149] Here, x and y are the horizontal and vertical positions of the sub-block, respectively, and w is a predetermined weighted coefficient.

[0150] An affine motion compensation prediction mode may include a number of modes of different methods of deriving the motion vectors of the top left and top right corner control points. Information indicating an affine motion compensation prediction mode (referred to as, for example, an affine flag) may be signaled at the CU level. Note that the signaling of information indicating the affine motion compensation prediction mode need not be performed at the CU level, and may be performed at another level (for example, at the sequence level, picture level, slice level, tile level, CTU level, or sub-block level).

[0151] (Prediction Controller)

Prediction controller 128 selects either the intra prediction signal (outputted from intra predictor 124) or the inter prediction signal (outputted from inter predictor 126), and outputs the selected prediction signal to subtractor 104 and adder 116.

[0152] As illustrated in FIG. 1, in various implementations, the prediction controller 128 may output prediction parameters, which are inputted to entropy encoder 110. Entropy encoder 110 may generate an encoded bitstream (or sequence) based on the prediction parameters, inputted from prediction controller 128, and the quantized coefficients, inputted from quantizer 108. The prediction parameters may be used by the decoder, which receives and decodes the encoded bitstream, to carry out the same prediction processing as performed in intra predictor 124, inter predictor 126, and prediction controller 128. The prediction parameters may include the selected prediction signal (e.g., motion vectors, prediction type or prediction mode employed in intra predictor 124 or inter predictor 126), or any index, flag, or value that is based on, or is indicative of, the prediction processing performed in intra predictor 124, inter predictor 126, and prediction controller 128.

[0153] FIG. 9B illustrates one example of a process for deriving a motion vector in a current picture in merge mode.

[0154] First, a prediction MV list is generated, in which prediction MV candidates are registered. Examples of prediction MV candidates include: spatially neighboring prediction MV, which are MVs of encoded blocks positioned in the spatial vicinity of the target block; temporally neighboring prediction MVs, which are MVs of blocks in encoded reference pictures that neighbor a block in the same location as the target block; a coupled prediction MV, which is an MV generated by combining the MV values of the spatially neighboring prediction MV and the temporally neighboring prediction MV; and a zero prediction MV, which is an MV whose value is zero.

[0155] Next, the MV of the target block is determined by selecting one prediction MV from among the plurality of prediction MVs registered in the prediction MV list.

[0156] Further, in a variable-length encoder, a merge_idx, which is a signal indicating which prediction MV is selected, is written and encoded into the stream.

[0157] Note that the prediction MVs registered in the prediction MV list illustrated in FIG. 9B constitute one example. The number of prediction MVs registered in the prediction MV list may be different from the number illustrated in FIG. 9B, and the prediction MVs registered in the prediction MV list may omit one or more of the types of prediction MVs given in the example in FIG. 9B, and the prediction MVs registered in the prediction MV list may include one or more types of prediction MVs in addition to and different from the types given in the example in FIG. 9B.

[0158] The final MV may be determined by performing DMVR (dynamic motion vector refreshing) processing (to be described later) by using the MV of the target block derived in merge mode.

[0159] FIG. 9C is a conceptual diagram illustrating an example of DMVR processing to determine an MV.

[0160] First, the most appropriate MV which is set for the current block (e.g., in merge mode) is considered to be the candidate MV. Then, according to candidate MV(L0), a reference pixel is identified in a first reference picture (L0) which is an encoded picture in L0 direction. Similarly, according to candidate MV(L1), a reference pixel is identified in a second reference picture (L1) which is an encoded picture in L1 direction. The reference pixels are then averaged to form a template.

[0161] Next, using the template, the surrounding regions of the candidate MVs of the first and second reference pictures (L0) and (L1) are searched, and the MV with the lowest cost is determined to be the final MV. The cost value may be calculated, for example, using the difference between each pixel value in the template and each pixel value in the regions searched, using the candidate MVs, etc.

[0162] Note that the configuration and operation of the processes described here are fundamentally the same in both the encoder side and the decoder side, to be described below.

[0163] Any processing other than the processing described above may be used, as long as the processing is capable of deriving the final MV by searching the surroundings of the candidate MV.

[0164] Next, a description is given of an example of a mode that generates a prediction image (a prediction) using LIC (local illumination compensation) processing.

[0165] FIG. 9D illustrates one example of a prediction image generation method using a luminance correction process performed by LIC processing.

[0166] First, from an encoded reference picture, an MV is derived to obtain a reference image corresponding to the current block.

[0167] Next, for the current block, information indicating how the luminance value changed between the reference picture and the current picture is obtained, based on the luminance pixel values of the encoded neighboring left reference region and the encoded neighboring upper reference region in the current picture, and based on the luminance pixel values in the same locations in the reference picture as specified by the MV. The information indicating how the luminance value changed is used to calculate a luminance correction parameter.

[0168] The prediction image for the current block is generated by performing a luminance correction process, which applies the luminance correction parameter on the reference image in the reference picture specified by the MV.

[0169] Note that the shape of the surrounding reference region(s) illustrated in FIG. 9D is just one example; the surrounding reference region may have a different shape.

[0170] Furthermore, although a prediction image is generated from a single reference picture in this example, in cases in which a prediction image is generated from a plurality of reference pictures, the prediction image may be generated after performing a luminance correction process, as described above, on the reference images obtained from the reference pictures.

[0171] One example of a method for determining whether to implement LIC processing is using an lic_flag, which is a signal that indicates whether to implement LIC processing. As one specific example, the encoder determines whether the current block belongs to a region of luminance change. The encoder sets the lic_flag to a value of "1" when the block belongs to a region of luminance change, and implements LIC processing when encoding. The encoder sets the lic_flag to a value of "0" when the block does not belong to a region of luminance change, and performs encoding implementing LIC processing. The decoder may switch between implementing LIC processing or not by decoding the lic_flag written in the stream and performing the decoding in accordance with the flag value.

[0172] One example of a different method of determining whether to implement LIC processing includes discerning whether LIC processing was determined to be implemented for a surrounding block. In one specific example, when merge mode is used on the current block, it is determined whether LIC processing was applied in the encoding of the surrounding encoded block, which was selected when deriving the MV in merge mode. Then, the determination is used to further determine whether to implement LIC processing or not for the current block. Note that in this example also, the same applies to the processing performed on the decoder side.

[0173] (Decoder)

Next, a decoder capable of decoding an encoded signal (encoded bitstream) output from encoder 100 will be described. FIG. 10 is a block diagram illustrating a functional configuration of decoder 200 according to an embodiment. Decoder 200 is a moving picture decoder that decodes a moving picture block by block.

[0174] As illustrated in FIG. 10, decoder 200 includes entropy decoder 202, inverse quantizer 204, inverse transformer 206, adder 208, block memory 210, loop filter 212, frame memory 214, intra predictor 216, inter predictor 218, and prediction controller 220.

[0175] Decoder 200 is realized as, for example, a generic processor and memory. In this case, when a software program stored in the memory is executed by the processor, the processor functions as entropy decoder 202, inverse quantizer 204, inverse transformer 206, adder 208, loop filter 212, intra predictor 216, inter predictor 218, and prediction controller 220. Alternatively, decoder 200 may be realized as one or more dedicated electronic circuits corresponding to entropy decoder 202, inverse quantizer 204, inverse transformer 206, adder 208, loop filter 212, intra predictor 216, inter predictor 218, and prediction controller 220.

[0176] Hereinafter, each component included in decoder 200 will be described.

[0177] (Entropy Decoder)

Entropy decoder 202 entropy decodes an encoded bitstream. More specifically, for example, entropy decoder 202 arithmetic decodes an encoded bitstream into a binary signal. Entropy decoder 202 then debinarizes the binary signal. Entropy decoder 202 outputs quantized coefficients of each block to inverse quantizer 204. Entropy decoder 202 may also output the prediction parameters, which may be included in the encoded bitstream (see FIG. 1), to intra predictor 216, inter predictor 218, and prediction controller 220 so that they can carry out the same prediction processing as performed on the encoder side in intra predictor 124, inter predictor 126, and prediction controller 128.

[0178] (Inverse Quantizer)

Inverse quantizer 204 inverse quantizes quantized coefficients of a block to be decoded (hereinafter referred to as a current block), which are inputted from entropy decoder 202. More specifically, inverse quantizer 204 inverse quantizes quantized coefficients of the current block based on quantization parameters corresponding to the quantized coefficients. Inverse quantizer 204 then outputs the inverse quantized coefficients (i.e., transform coefficients) of the current block to inverse transformer 206.

[0179] (Inverse Transformer)

Inverse transformer 206 restores prediction errors (residuals) by inverse transforming transform coefficients, which are inputted from inverse quantizer 204.

[0180] For example, when information parsed from an encoded bitstream indicates application of EMT or AMT (for example, when the AMT flag is set to true), inverse transformer 206 inverse transforms the transform coefficients of the current block based on information indicating the parsed transform type.

[0181] Moreover, for example, when information parsed from an encoded bitstream indicates application of NSST, inverse transformer 206 applies a secondary inverse transform to the transform coefficients.

[0182] (Adder)

Adder 208 reconstructs the current block by summing prediction errors, which are inputted from inverse transformer 206, and prediction samples, which is an input from prediction controller 220. Adder 208 then outputs the reconstructed block to block memory 210 and loop filter 212.

[0183] (Block Memory)

Block memory 210 is storage for storing blocks in a picture to be decoded (hereinafter referred to as a current picture) for reference in intra prediction. More specifically, block memory 210 stores reconstructed blocks output from adder 208.

[0184] (Loop Filter)

Loop filter 212 applies a loop filter to blocks reconstructed by adder 208, and outputs the filtered reconstructed blocks to frame memory 214 and, for example, to a display device.

[0185] When information indicating the enabling or disabling of ALF parsed from an encoded bitstream indicates enabled, one filter from among a plurality of filters is selected based on direction and activity of local gradients, and the selected filter is applied to the reconstructed block.

[0186] (Frame Memory)

Frame memory 214 is storage for storing reference pictures used in inter prediction, and is also referred to as a frame buffer. More specifically, frame memory 214 stores reconstructed blocks filtered by loop filter 212.

[0187] (Intra Predictor)

Intra predictor 216 generates a prediction signal (intra prediction signal) by intra prediction with reference to a block or blocks in the current picture as stored in block memory 210. More specifically, intra predictor 216 generates an intra prediction signal by intra prediction with reference to samples (for example, luma and/or chroma values) of a block or blocks neighboring the current block, and then outputs the intra prediction signal to prediction controller 220.

[0188] Note that when an intra prediction mode in which a chroma block is intra predicted from a luma block is selected, intra predictor 216 may predict the chroma component of the current block based on the luma component of the current block.

[0189] Moreover, when information indicating the application of PDPC is parsed from an encoded bitstream (in the prediction parameters outputted from entropy decoder 202, for example), intra predictor 216 corrects post-intra-prediction pixel values based on horizontal/vertical reference pixel gradients.

[0190] (Inter Predictor)

Inter predictor 218 predicts the current block with reference to a reference picture stored in frame memory 214. Inter prediction is performed per current block or per sub-block (for example, per 4×4 block) in the current block. For example, inter predictor 218 generates an inter prediction signal of the current block or sub-block based on motion compensation using motion information (for example, a motion vector) parsed from an encoded bitstream (in the prediction parameters outputted from entropy decoder 202, for example), and outputs the inter prediction signal to prediction controller 220.

[0191] When the information parsed from the encoded bitstream indicates application of OBMC mode, inter predictor 218 generates the inter prediction signal using motion information for a neighboring block in addition to motion information for the current block obtained from motion estimation.

[0192] Moreover, when the information parsed from the encoded bitstream indicates application of FRUC mode, inter predictor 218 derives motion information by performing motion estimation in accordance with the pattern matching method (bilateral matching or template matching) parsed from the encoded bitstream. Inter predictor 218 then performs motion compensation (prediction) using the derived motion information.

[0193] Moreover, when BIO mode is to be applied, inter predictor 218 derives a motion vector based on a model assuming uniform linear motion. Further, when the information parsed from the encoded bitstream indicates that affine motion compensation prediction mode is to be applied, inter predictor 218 derives a motion vector of each sub-block based on motion vectors of neighboring blocks.

[0194] (Prediction Controller)

Prediction controller 220 selects either the intra prediction signal or the inter prediction signal, and outputs the selected prediction signal to adder 208. In general, the configuration, functions and operations of prediction controller 220, inter predictor 218 and intra predictor 216 on the decoder side may correspond to the configuration, functions and operations of prediction controller 128, inter predictor 126 and intra predictor 124 on the encoder side.

[0195] (Non-rectangular Partitioning)

In prediction controller 128 coupled to intra predictor 124 and inter predictor 126 on the encoder side (see FIG. 1) as well as in prediction controller 220 coupled to intra predictor 216 and inter predictor 218 on the decoder side (see FIG. 10), heretofore partitions (or variable size blocks or sub-blocks) obtained from splitting each block, for which motion information (e.g., motion vectors) are obtained, are invariably rectangular, as shown in FIG. 2. The inventors have discovered that generating partitions having a non-rectangular shape, such as a triangular shape, leads to an improvement in image quality and encoding efficiency depending on the content of an image in a picture in various implementations. Below, various embodiments will be described, in which at least one partition split from an image block for the purpose of prediction has a non-rectangular shape. Note that these embodiments are equally applicable on the encoder side (prediction controller 128 coupled to intra predictor 124 and inter predictor 126) and on the decoder side (prediction controller 220 coupled to intra predictor 216 and inter predictor 218), and may be implemented in the encoder of FIG. 1 or the like, or in the decoder of FIG. 10 or the like.

[0196] FIG. 11 is a flow chart illustrating one example of a process of splitting an image block into partitions including at least a first partition having a non-rectangular shape (e.g., a triangle) and a second partition, and performing further processing including encoding (or decoding) the image block as a reconstructed combination of the first and second partitions.

[0197] In step S1001, an image block is split into partitions including a first partition having a non-rectangular shape and a second partition, which may or may not have a non-rectangular shape. For example, as shown in FIG. 12, an image block may be split from a top-left corner of the image block to a bottom-right corner of the image block to create a first partition and a second partition both having a non-rectangular shape (e.g., a triangle), or an image block may be split from a top-right corner of the image block to a bottom-left corner of the image block to create a first partition and a second partition both having a non-rectangular shape (e.g., a triangle). Various examples of the non-rectangular partitioning will be described below in reference to FIGS. 12 and 17-19.

[0198] In step S1002, the process predicts a first motion vector for the first partition and predicts a second motion vector for the second partition. For example, the predicting of the first and second motion vectors may include selecting the first motion vector from a first set of motion vector candidates and selecting the second motion vector from a second set of motion vector candidates.

[0199] In step S1003, a motion compensation process is performed to obtain the first partition using the first motion vector, which is derived in step S1002 above, and to obtain the second partition using the second motion vector, which is derived in step

S1002 above.

[0200] In step S1004, a prediction process is performed for the image block as a (reconstructed) combination of the first partition and the second partition. The prediction process may include a boundary smoothing process to smooth out the boundary between the first partition and the second partition. For example, the boundary smoothing process may involve weighting first values of boundary pixels predicted based on the first partition and second values of the boundary pixels predicted based on the second partition. Various implementations of the boundary smoothing process will be described below in reference to FIGS. 13, 14, 20 and 21A-21D.

[0201] In step S1005, the process encodes or decodes the image block using one or more parameters including a partition parameter indicative of the splitting of the image block into the first partition having a non-rectangular shape and the second partition. As summarized in a table of FIG. 15, for example, the partition parameter ("the first index value") may jointly encode, for example, a split direction applied in the splitting (e.g., from top-left to bottom-right or from top-right to bottom-left as shown in FIG. 12) and the first and second motion vectors derived in step S1002 above. Details of such partition syntax operation involving the one or more parameters including the partition parameter will be described in detail below in reference to FIGS. 15, 16 and 22-25.

[0202] FIG. 17 is a flowchart illustrating a process 2000 of splitting an image block. In step S2001, the process splits an image into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, which may or may not have a non-rectangular shape. As shown in FIG. 12, an image block may be split into a first partition having a triangle shape and a second partition also having a triangle shape. There are numerous other examples in which an image block is split into a plurality of partitions including a first partition and a second partition of which at least the first partition has a non-rectangular shape. The non-rectangular shape may be a triangle, a trapezoid, and a polygon with at least five sides and angles.

[0203] For example, as shown in FIG. 18, an image block may be split into two triangular shape partitions; an image block may be split into more than two triangular shape partitions (e.g., three triangular shape partitions); an image block may be split into a combination of triangular shape partition(s) and rectangular shape partition(s); or an image block may be split into a combination of triangle shape partition(s) and polygon shape partition(s).

[0204] As further shown in FIG. 19, an image block may be split into an L-shaped (polygon shape) partition and a rectangular shape partition; an image block may be split into a pentagon (polygon) shape partition and a triangular shape partition; an image block may be split into a hexagon (polygon) shape partition and a pentagon (polygon) shape partition; or an image block may be split into multiple polygon shape partitions.

[0205] Referring back to FIG. 17, in step S2002, the process predicts a first motion vector for the first partition, for example by selecting the first partition from a first set of motion vector candidates, and predicts a second motion vector for the second partition, for example by selecting the second partition from a second set of motion vector candidates. For example, the first set of motion vector candidates may include motion vectors of partitions neighboring the first partition, and the second set of motion vector candidates may include motion vectors of partitions neighboring the second partition. The neighboring partitions may be one or both of spatially neighboring partitions and temporary neighboring partitions. Some examples of the spatially neighboring partitions include a partition located at the left, bottom-left, bottom, bottom-right, right, top-right, top, or top-left of the partition that is being processed. Examples of the temporary neighboring partitions are co-located partitions in the reference pictures of the image block.

[0206] In various implementations, the partitions neighboring the first partition and the partitions neighboring the second partition may be outside of the image block from which the first partition and the second partition are split. The first set of motion vector candidates may be the same as, or different from, the second set of motion vector candidates. Further, at least one of the first set of motion vector candidates and the second set of motion vector candidates may be the same as another, third set of motion vector candidates prepared for the image block.

[0207] In some implementations, in step S2002, in response to determining that the second partition, similar to the first partition, too has a non-rectangular shape (e.g., a triangle), the process 2000 creates the second set of motion vector candidates (for the non-rectangular shape second partition) that includes motion vectors of partitions neighboring the second partition exclusive of the first partition (i.e., exclusive of the motion vector of the first partition). On the other hand, in response to determining that the second partition, unlike the first partition, has a rectangular shape, the process 2000 creates the second set of motion vector candidates (for the rectangular shape second partition) that includes motion vectors of partitions neighboring the second partition inclusive of the first partition.

[0208] In step S2003, the process encodes or decodes the first partition using the first motion vector derived in step S2002 above, and encodes or decodes the second partition using the second motion vector derived in step S2002 above.

[0209] An image block splitting process, like the process 2000 of FIG. 17, may be performed by an image encoder, as shown in FIG. 1 for example, which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition (step S2001); predicting a first motion vector for the first partition and a second motion vector for the second partition (step S2002); and encoding the first partition using the first motion vector and the second partition using the second motion vector (step S2003).

[0210] According to another embodiment, as shown in FIG. 1, an image encoder is provided including: a splitter 102 which, in operation, receives and splits an original picture into blocks; an adder 104 which, in operation, receives the blocks from the splitter and predictions from a prediction controller 128, and subtracts each prediction from its corresponding block to output a residual; a transformer 106 which, in operation, performs a transform on the residuals outputted from the adder 104 to output transform coefficients; a quantizer 108 which, in operation, quantizes the transform coefficients to generate quantized transform coefficients; an entropy encoder 110 which, in operation, encodes the quantized transform coefficients to generate a bitstream; and the prediction controller 128 coupled to an inter predictor 126, an intra predictor 124, and a memory 118, 122, wherein the inter predictor 126, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor 124, in operation, generates a prediction of a current block based on an encoded reference block in a current picture. The prediction controller 128, in operation, splits the blocks into a plurality of partitions including a first partition having a non-rectangular shape and a second partition (FIG. 17, step S2001); predicts a first motion vector for the first partition and a second motion vector for the second partition (step S2002); and encodes the first partition using the first motion vector and the second partition using the second motion vector (step S2003).

[0211] According to another embodiment, an image decoder, as shown in FIG. 10 for example, is provided which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition (FIG. 17, step S2001); predicting a first motion vector for the first partition and a second motion vector for the second partition (step S2002); and decoding the first partition using the first motion vector and the second partition using the second motion vector (step S2003).

[0212] According to a further embodiment, an image decoder as shown in FIG. 10 is provided including: an entropy decoder 202 which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients; an inverse quantizer 204 and transformer 206 which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals; an adder 208 which, in operation, adds the residuals outputted from the inverse quantizer 204 and transformer 206 and predictions outputted from a prediction controller 220 to reconstruct blocks; and the prediction controller 220 coupled to an inter predictor 218, an intra predictor 216, and a memory 210, 214, wherein the inter predictor 218, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor 216, in operation, generates a prediction of a current block based on an decoded reference block in a current picture. The prediction controller 220, in operation, splits an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition (FIG. 17, step S2001); predicts a first motion vector for the first partition and a second motion vector for the second partition (step S2002); and decodes the first partition using the first motion vector and the second partition using the second motion vector (step S2003).

[0213] (Boundary Smoothing)

As described above in FIG. 11, step S1004, according to various embodiments, performing a prediction process for the image block as a (reconstructed) combination of the first partition having a non-rectangular shape and the second partition may involve application of a boundary smoothing process along the boundary between the first partition and the second partition.

[0214] For example, FIG. 21B illustrates one example of a boundary smoothing process involving weighting first values of boundary pixels, which are first-predicted based on the first partition, and second values of the boundary pixels, which are second-predicted based on the second partition.

[0215] FIG. 20 is a flowchart illustrating an overall boundary smoothing process 3000 involving weighting first values of boundary pixels first-predicted based on the first partition and second values of the boundary pixels second-predicted based on the second partition, according to one embodiment. In step S3001, an image block is split into a first partition and a second partition along a boundary wherein at least the first partition has a non-rectangular shape, as shown in FIG. 21A or in FIGS. 12, 18 and 19 described above.

[0216] In step S3002, first values (e.g., color, luminance, transparency, etc.) of a set of pixels ("boundary pixels" in FIG. 21A) of the first partition along the boundary are first-predicted, wherein the first values are first-predicted using information of the first partition. In step S3003, second values of the (same) set of pixels of the first partition along the boundary are second-predicted, wherein the second values are second-predicted using information of the second partition. In some implementation, at least one of the first-predicting and the second-predicting is an inter prediction process that predicts the first values and the second values based on a reference partition in an encoded reference picture. Referring to FIG. 21D, in some implementations, the prediction process predicts first values of all pixels of the first partition ("the first set of samples") including the set of pixels over which the first partition and the second partition overlap, and predicts second values of only the set of pixels ("the second set of samples") over which the first and second partitions overlap. In another implementation, at least one of the first-predicting and the second-predicting is an intra prediction process that predicts the first values and the second values based on an encoded reference partition in a current picture. In some implementations, a prediction method used in the first-predicting is different from a prediction method used in the second-predicting. For example, the first-predicting may include an inter prediction process and the second-predicting may include an intra prediction process. The information used to first-predict the first values or to second-predict the second values may be motion vectors, intra-prediction directions, etc. of the first or second partition.

[0217] In step S3004, the first values, predicted using the first partition, and the second values, predicted using the second partition, are weighted. In step S3005, the first partition is encoded or decoded using the weighted first and second values.

[0218] FIG. 21B illustrates an example of a boundary smoothing operation wherein the first partition and the second partition overlap over five pixels (at a maximum) of each row or each column. That is, the number of the set of pixels of each row or each column, for which the first values are predicted based on the first partition and the second values are predicted based on the second partition, are five at a maximum. FIG. 21C illustrates another example of a boundary smoothing operation wherein the first partition and the second partition overlap over three pixels (at a maximum) of each row or each column. That is, the number of the set of pixels of each row or each column, for which the first values are predicted based on the first partition and the second values are predicted based on the second partition, are three at a maximum.

[0219] FIG. 13 illustrates another example of boundary smoothing operation wherein the first partition and the second partition overlap over four pixels (at a maximum) of each row or each column. That is, the number of the set of pixels of each row or each column, for which the first values are predicted based on the first partition and the second values are predicted based on the second partition, are four at a maximum. In the illustrated example, weights of 1/8, 1/4, 3/4, and 7/8 may be applied to the first values of the four pixels in the set, respectively, and weights of 7/8, 3/4, 1/4, and 1/8 may be applied to the second values of the four pixels in the set, respectively.

[0220] FIG. 14 illustrate further examples of a boundary smoothing operation wherein the first partition and the second partition overlap over zero pixels of each row or each column (i.e., they do not overlap), overlap over one pixel (at a maximum) of each row or each column, and overlap over two pixels (at a maximum) of each row or each column, respectively. In the example wherein the first and second partitions do not overlap, zero weights are applied. In the example wherein the first and second partitions overlap over one pixel of each row or each column, a weight of 1/2 may be applied to the first values of the pixels in the set predicted based on the first partition, and a weight of 1/2 may be applied to the second values of the pixels in the set predicted based on the second partition. In the example wherein the first and second partitions overlap over two pixels of each row or each column, weights of 1/3 and 2/3 may be applied to the first values of the two pixels in the set predicted based on the first partition, respectively, and weights of 2/3 and 1/3 may be applied to the second values of the two pixels in the set predicted based on the second partition, respectively.

[0221] According to the embodiments described above, the number of pixels in the set over which the first partition and the second partition overlap is an integer. In other implementations, the number of overlapping pixels in the set may be non-integer and may be fractional, for example. Also, the weights applied to the first and second values of the set of pixels may be fractional or integer depending on each application.

[0222] A boundary smoothing process, like the process 3000 of FIG. 20, may be performed by an image encoder, as shown in FIG. 1 for example, which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block (FIG. 20, step S3001). The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition (step S3002); second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition (step S3003); weighting the first values and the second values (step S3004); and encoding the first partition using the weighted first values and the weighted second values (step S3005).

[0223] According to another embodiment, as shown in FIG. 1, an image encoder is provided including: a splitter 102 which, in operation, receives and splits an original picture into blocks; an adder 104 which, in operation, receives the blocks from the splitter and predictions from a prediction controller 128, and subtracts each prediction from its corresponding block to output a residual; a transformer 106 which, in operation, performs a transform on the residuals outputted from the adder 104 to output transform coefficients; a quantizer 108 which, in operation, quantizes the transform coefficients to generate quantized transform coefficients; an entropy encoder 110 which, in operation, encodes the quantized transform coefficients to generate a bitstream; and the prediction controller 128 coupled to an inter predictor 126, an intra predictor 124, and a memory 118, 122, wherein the inter predictor 126, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor 124, in operation, generates a prediction of a current block based on an encoded reference block in a current picture. The prediction controller 128, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block (FIG. 20, step S3001). The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition (step S3002); second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition (step S3003); weighting the first values and the second values (step S3004); and encoding the first partition using the weighted first values and the weighted second values (step S3005).

[0224] According to another embodiment, an image decoder is provided, as shown in FIG. 10 for example, which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block (FIG. 20, steps S3001). The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition (step S3002); second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition (step S3003); weighting the first values and the second values (step S3004); and decoding the first partition using the weighted first values and the weighted second values (step S3005).

[0225] According to another embodiment, an image decoder as shown in FIG 10 is provided including: an entropy decoder 202 which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients; an inverse quantizer 204 and transformer 206 which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals; an adder 208which, in operation, adds the residuals outputted from the inverse quantizer 204 and transformer 206 and predictions outputted from a prediction controller 220 to reconstruct blocks; and the prediction controller 220 coupled to an inter predictor 218, an intra predictor 216, and a memory 210, 214, wherein the inter predictor 218, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor 216, in operation, generates a prediction of a current block based on an decoded reference block in a current picture. The prediction controller 220, in operation, performs a boundary smoothing operation along a boundary between a first partition having a non-rectangular shape and a second partition that are split from an image block. (FIG. 20, step S3001) The boundary smoothing operation includes: first-predicting first values of a set of pixels of the first partition along the boundary, using information of the first partition (step S3002); second-predicting second values of the set of pixels of the first partition along the boundary, using information of the second partition (step S3003); weighting the first values and the second values (step S3004); and decoding the first partition using the weighted first values and the weighted second values (step S3005).

[0226] (Entropy Encoding and Decoding using Partition Parameter Syntax)

As described in FIG. 11, step S1005, according to various embodiments, the image block split into a first partition having a non-rectangular shape and a second partition may be encoded or decoded using one or more parameters including a partition parameter indicative of the non-rectangular splitting of the image block. In various embodiments, such partition parameter may jointly encode, for example, a split direction applied to the splitting (e.g., from top-left to bottom-right or from top-right to bottom-left, see FIG. 12) and the first and second motion vectors predicted in step S1002, as will be more fully described below.

[0227] FIG. 15 is a table of sample partition parameters ("the first index value") and sets of information jointly encoded by the partition parameters, respectively. The partition parameters ("the first index values") range from 0 to 6 and jointly encode: the direction of splitting an image block into a first partition and a second partition both of which are triangles (see FIG. 12), the first motion vector predicted for the first partition (FIG. 11, step S1002), and the second motion vector predicted for the second partition (FIG. 11, step S1002). Specifically, the partition parameter 0 encodes the split direction is from top-left corner to bottom-right corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition.

[0228] The partition parameter 1 encodes the split direction is from top-right corner to bottom-left corner, the first motion vector is the "1st" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "2nd" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 2 encodes the split direction is from top-right corner to bottom-left corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 3 encodes the split direction is from top-left corner to bottom-right corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "2nd" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 4 encodes the split direction is from top-right corner to bottom-left corner, the first motion vector is the

"2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "3rd" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 5 encodes the split direction is from top-left corner to bottom-right corner, the first motion vector is the "3rd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 6 encodes the split direction is from top-left corner to bottom-right corner, the first motion vector is the "4th" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition.

[0229] FIG. 22 is a flowchart illustrating a method 4000 performed on the encoder side. In step S4001, the process splits an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, based on a partition parameter indicative of the splitting. For example, as shown in FIG. 15 described above, the partition parameter may indicate the direction of splitting an image block (e.g., from top-right corner to bottom-left corner or from top-left corner to bottom-right corner). In step S4002, the process encodes the first partition and the second partition. In step S4003, the process writes one or more parameters including the partition parameter into a bit stream, which the decoder side can receive and decode to obtain the one or more parameters to perform the same prediction process (as performed on the encoder side) for the first and second partitions on the decoder side. The one or more parameters including the partition parameter may jointly or separately encode various pieces of information such as the non-rectangular shape of the first partition, the shape of the second partition, the split direction used to split an image block to obtain the first and second partitions, the first motion vector of the first partition, the second motion vector of the second partition, etc.

[0230] FIG. 23 is a flowchart illustrating a method 5000 performed on the decoder side. In step S5001, the process parses one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition. The one or more parameters including the partition parameter parsed out of the bitstream may jointly or separately encode various pieces of information needed for the decoder side to perform the same prediction process as performed on the encoder side, such as the non-rectangular shape of the first partition, the shape of the second partition, the split direction used to split an image block to obtain the first and second partitions, the first motion vector of the first partition, the second motion vector of the second partition, etc. In step S5002, the process 5000 splits the image block into the plurality of partitions based on the partition parameter parsed out of the bitstream. In step S5003, the process decodes the first partition and the second partition, as split from the image block.

[0231] FIG. 24 is a table of sample partition parameters ("the first index value") and sets of information jointly encoded by the partition parameters, respectively, similar in nature to the sample table described above in FIG. 15. In FIG. 24, the partition parameters ("the first index values") range from 0 to 6 and jointly encode: the shape of the first and second partitions split from an image block, the direction of splitting an image block into the first and second partitions, the first motion vector predicted for the first partition (FIG. 11, step S1002), and the second motion vector predicted for the second partition (FIG. 11, step S1002). Specifically, the partition parameter 0 encodes that neither of the first and second partitions has a triangular shape, and thus the split direction information is "N/A", the first motion vector information is "N/A", and the second motion vector information is "N/A".

[0232] The partition parameter 1 encodes the first and second partitions are triangles, the split direction is from top-left corner to bottom-right corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 2 encodes the first and second partitions are triangles, the split direction is from top-right corner to bottom-left corner, the first motion vector is the "1st" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "2nd" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 3 encodes the first and second partitions are triangles, the split direction is from top-right corner to bottom-left corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 4 encodes the first and second partitions are triangles, the split direction is from top-left corner to bottom-right corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "2nd" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 5 encodes the first and second partitions are triangles, the split direction is from top-right corner to bottom-left corner, the first motion vector is the "2nd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "3rd" motion vector listed in the second set of motion vector candidates for the second partition. The partition parameter 6 encodes the first and second partitions are triangles, the split direction is from top-left corner to bottom-right corner, the first motion vector is the "3rd" motion vector listed in the first set of motion vector candidates for the first partition, and the second motion vector is the "1st" motion vector listed in the second set of motion vector candidates for the second partition.

[0233] According to some implementations, the partition parameters (index values) may be binarized pursuant to a binarization scheme, which is selected depending on a value of at least one or the one or more parameters. FIG. 16 illustrates a sample binarization scheme of binarizing the index values (the partition parameter values).

[0234] FIG. 25 is a table of sample combinations of a first parameter and a second parameter, wherein one of which is a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition. In this example, the partition parameter may be used to indicate splitting of an image block without jointly encoding other information, which is encoded by one or more of the other parameters.

[0235] In the first example in FIG. 25, the first parameter is used to indicate an image block size, and the second parameter is used as the partition parameter (a flag) to indicate that at least one of a plurality of partitions split from an image block has a triangular shape. Such combination of the first and second parameters may be used to indicate, for example, 1) when the image block size is larger than 64x64, there is no triangular shape partition, or 2) when the ratio of width and height of an image block is larger than 4 (e.g., 64x4), there is no triangular shape partition.

[0236] In the second example of FIG. 25, the first parameter is used to indicate a prediction mode, and the second parameter is used as the partition parameter (a flag) to indicate that at least one of a plurality of partitions split from an image block has a triangular shape. Such combination of the first and second parameters may be used to indicate, for example, 1) when an image block is coded in intra mode, there is no triangular partition.

[0237] In the third example of FIG. 25, the first parameter is used as the partition parameter (a flag) to indicate that at least one of a plurality of partitions split from an image block has a triangular shape, and the second parameter is used to indicate a prediction mode. Such combination of the first and second parameters may be used to indicate, for example, 1) when at least one of the plurality of partitions split from an image block has a triangular shape, the image block must be inter coded.

[0238] In the fourth example of FIG. 25, the first parameter indicates the motion vector of a neighboring block, and the second parameter is used as the partition parameter which indicates the direction of splitting an image block into two triangles. Such combination of the first and second parameters may be used to indicate, for example, 1) when the motion vector of a neighboring block is a diagonal direction, the direction of splitting the image block into two triangles is from top-left corner to bottom-right corner.

[0239] In the fifth example of FIG. 25, the first parameter indicates the intra prediction direction of a neighboring block, and the second parameter is used as the partition parameter which indicates the direction of splitting an image block into two triangles. Such combination of the first and second parameters may be used to indicate, for example, 1) when the intra prediction direction of a neighboring block is an inverse-diagonal direction, the direction of splitting the image block into two triangles is from top-right corner to bottom-left corner.

[0240] It should be understood that the tables of one or more parameters including the partition parameter and what information is jointly or separately encoded, as shown in FIGS. 15, 24, and 25, are presented as examples only and numerous other ways of encoding, jointly or separately, various information as part of the partition syntax operation described above are within the scope of the present disclosure. For example, the partition parameter may indicate the first partition is a triangle, a trapezoid, or a polygon with at least five sides and angles. The partition parameter may indicate the second partition has a non-rectangular shape, such as a triangle, a trapezoid, and a polygon with at least five sides and angles. The partition parameter may indicate one or more pieces of information about the splitting, such as the non-rectangular shape of the first partition, the shape of the second partition (which may be non-rectangular or rectangular), the split direction applied to split an image block into a plurality of partitions (e.g., from a top-left corner of the image block to a bottom-right corner thereof, and from a top-right corner of the image block to a bottom-left corner thereof). The partition parameter may jointly encode further information such as the first motion vector of the first partition, the second motion vector of the second partition, image block size, prediction mode, the motion vector of a neighboring block, the intra prediction direction of a neighboring block, etc. Alternatively, any of the further information may be separately encoded by one or more parameters other than the partition parameter.

[0241] A partition syntax operation, like the process 4000 of FIG. 22, may be performed by an image encoder, as shown in FIG. 1 for example, which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a partition syntax operation including: splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition based on a partition parameter indicative of the splitting (FIG. 22, step S4001); encoding the first partition and the second partition (S4002); and writing one or more parameters including the partition parameter into a bitstream (S4003).

[0242] According to another embodiment, as shown in FIG. 1, an image encoder is provided including: a splitter 102 which, in operation, receives and splits an original picture into blocks; an adder 104 which, in operation, receives the blocks from the splitter and predictions from a prediction controller 128, and subtracts each prediction from its corresponding block to output a residual; a transformer 106 which, in operation, performs a transform on the residuals outputted from the adder 104 to output transform coefficients; a quantizer 108 which, in operation, quantizes the transform coefficients to generate quantized transform coefficients; an entropy encoder 110 which, in operation, encodes the quantized transform coefficients to generate a bitstream; and the prediction controller 128 coupled to an inter predictor 126, an intra predictor 124, and a memory 118, 122, wherein the inter predictor 126, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor 124, in operation, generates a prediction of a current block based on an encoded reference block in a current picture. The prediction controller 128, in operation, splits an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition based on a partition parameter indicative of the splitting (FIG. 22, step S4001), and encodes the first partition and the second partition (step S4002). The entropy encoder 110, in operation, writes one or more parameters including the partition parameter into a bitstream (step S4003).

[0243] According to another embodiment, an image decoder is provided, as shown in FIG. 10 for example, which includes circuitry and a memory coupled to the circuitry. The circuitry, in operation, performs a partition syntax operation including: parsing one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition (FIG. 23, step S5001); splitting the image block into the plurality of partitions based on the partition parameter (S5002); and decoding the first partition and the second partition (S5003).

[0244] According to a further embodiment, an image decoder as shown in FIG. 10 is provided including: an entropy decoder 202which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients; an inverse quantizer 204 and transformer 206 which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals; an adder 208 which, in operation, adds the residuals outputted from the inverse quantizer 204 and transformer 206 and predictions outputted from a prediction controller 220 to reconstruct blocks; and the prediction controller 220 coupled to an inter predictor 218, an intra predictor 216, and a memory 210, 214, wherein the inter predictor 218, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor 216, in operation, generates a prediction of a current block based on an decoded reference block in a current picture. The entropy decoder 202, in operation: parses one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition (FIG. 23, step S5001); splits the image block into the plurality of partitions based on the partition parameter (S5002); and decodes the first partition and the second partition (S5003) in cooperation with the prediction controller 220 in some implementations.

[0245] According to other examples, an inter predictor may perform the following process.

[0246] All motion vector candidates included in the first set of motion vector candidates may be uni-prediction motion vectors. That is, the inter predictor may determine only uni-prediction motion vectors as motion vector candidates in the first set of motion vector candidates.

[0247] The inter predictor may select only uni-prediction motion vector candidates from the first set of motion vector candidates.

[0248] Only uni-prediction motion vector may be used to predict a small block. The bi-prediction motion vector may be used to predict a big block. As one example, the predicting process may include judging a size of the image block. When the size of the image block is judged to be larger than a threshold, the predicting may include selecting the first motion vector from a first set of motion vector candidates, and the first set of motion vector candidates may contain uni-prediction and/or bi-prediction motion vectors. When the size of the image block is judged to be not larger than the threshold, the predicting may include selecting the first motion vector from a first set of motion vector candidates, and the first set of motion vector candidates may contain only uni-prediction motion vectors.

[0249] (Implementations and Applications)

As described in each of the above embodiments, each functional or operational block can typically be realized as an MPU (micro processing unit) and memory, for example. Moreover, processes performed by each of the functional blocks may be realized as a program execution unit, such as a processor which reads and executes software (a program) recorded on a recording medium such as ROM. The software may be distributed. The software may be recorded on a variety of recording media such as semiconductor memory. Note that each functional block can also be realized as hardware (dedicated circuit).

[0250] The processing described in each of the embodiments may be realized via integrated processing using a single apparatus (system), and, alternatively, may be realized via decentralized processing using a plurality of apparatuses. Moreover, the processor that executes the above-described program may be a single processor or a plurality of processors. In other words, integrated processing may be performed, and, alternatively, decentralized processing may be performed.

[0251] Embodiments of the present disclosure are not limited to the above exemplary embodiments; various modifications may be made to the exemplary embodiments, the results of which are also included within the scope of the embodiments of the present disclosure.

[0252] Next, application examples of the moving picture encoding method (image encoding method) and the moving picture decoding method (image decoding method) described in each of the above embodiments will be described, as well as various systems that implement the application examples. Such a system may be characterized as including an image encoder that employs the image encoding method, an image decoder that employs the image decoding method, or an image encoder-decoder that includes both the image encoder and the image decoder. Other configurations of such a system may be modified on a case-by-case basis.

[0253] (Usage Examples)

FIG. 26 illustrates an overall configuration of content providing system ex100 suitable for implementing a content distribution service. The area in which the communication service is provided is divided into cells of desired sizes, and base stations ex106, ex107, ex108, ex109, and ex110, which are fixed wireless stations in the illustrated example, are located in respective cells.

[0254] In content providing system ex100, devices including computer ex111, gaming device ex112, camera ex113, home appliance ex114, and smartphone ex115 are connected to internet ex101 via internet service provider ex102 or communications network ex104 and base stations ex106 through ex110. Content providing system ex100 may combine and connect any combination of the above devices. In various implementations, the devices may be directly or indirectly connected together via a telephone network or near field communication, rather than via base stations ex106 through ex110. Further, streaming server ex103 may be connected to devices including computer ex111, gaming device ex112, camera ex113, home appliance ex114, and smartphone ex115 via, for example, internet ex101. Streaming server ex103 may also be connected to, for example, a terminal in a hotspot in airplane ex117 via satellite ex116.

[0255] Note that instead of base stations ex106 through ex110, wireless access points or hotspots may be used. Streaming server ex103 may be connected to communications network ex104 directly instead of via internet ex101 or internet service provider ex102, and may be connected to airplane ex117 directly instead of via satellite ex116.

[0256] Camera ex113 is a device capable of capturing still images and video, such as a digital camera. Smartphone ex115 is a smartphone device, cellular phone, or personal handyphone system (PHS) phone that can operate under the mobile communications system standards of the 2G, 3G, 3.9G, and 4G systems, as well as the next-generation 5G system.

[0257] Home appliance ex114 is, for example, a refrigerator or a device included in a home fuel cell cogeneration system.

[0258] In content providing system ex100, a terminal including an image and/or video capturing function is capable of, for example, live streaming by connecting to streaming server ex103 via, for example, base station ex106. When live streaming, a terminal (e.g., computer ex111, gaming device ex112, camera ex113, home appliance ex114, smartphone ex115, or airplane ex117) may perform the encoding processing described in the above embodiments on still-image or video content captured by a user via the terminal, may multiplex video data obtained via the encoding and audio data obtained by encoding audio corresponding to the video, and may transmit the obtained data to streaming server ex103. In other words, the terminal functions as the image encoder according to one aspect of the present disclosure.

[0259] Streaming server ex103 streams transmitted content data to clients that request the stream. Client examples include computer ex111, gaming device ex112, camera ex113, home appliance ex114, smartphone ex115, and terminals inside airplane ex117, which are capable of decoding the above-described encoded data. Devices that receive the streamed data decode and reproduce the received data. In other words, the devices may each function as the image decoder, according to one aspect of the present disclosure.

[0260] (Decentralized Processing)

Streaming server ex103 may be realized as a plurality of servers or computers between which tasks such as the processing, recording, and streaming of data are divided. For example, streaming server ex103 may be realized as a content delivery network (CDN) that streams content via a network connecting multiple edge servers located throughout the world. In a CDN, an edge server physically near the client is dynamically assigned to the client. Content is cached and streamed to the edge server to reduce load times. In the event of, for example, some type of error or change in connectivity due, for example, to a spike in traffic, it is possible to stream data stably at high speeds, since it is possible to avoid affected parts of the network by, for example, dividing the processing between a plurality of edge servers, or switching the streaming duties to a different edge server and continuing streaming.

[0261] Decentralization is not limited to just the division of processing for streaming; the encoding of the captured data may be divided between and performed by the terminals, on the server side, or both. In one example, in typical encoding, the processing is performed in two loops. The first loop is for detecting how complicated the image is on a frame-by-frame or scene-by-scene basis, or detecting the encoding load. The second loop is for processing that maintains image quality and improves encoding efficiency. For example, it is possible to reduce the processing load of the terminals and improve the quality and encoding efficiency of the content by having the terminals perform the first loop of the encoding and having the server side that received the content perform the second loop of the encoding. In such a case, upon receipt of a decoding request, it is possible for the encoded data resulting from the first loop performed by one terminal to be received and reproduced on another terminal in approximately real time. This makes it possible to realize smooth, real-time streaming.

[0262] In another example, camera ex113 or the like extracts a feature amount from an image, compresses data related to the feature amount as metadata, and transmits the compressed metadata to a server. For example, the server determines the significance of an object based on the feature amount and changes the quantization accuracy accordingly to perform compression suitable for the meaning (or content significance) of the image. Feature amount data is particularly effective in improving the precision and efficiency of motion vector prediction during the second compression pass performed by the server. Moreover, encoding that has a relatively low processing load, such as variable length coding (VLC), may be handled by the terminal, and encoding that has a relatively high processing load, such as context-adaptive binary arithmetic coding (CABAC), may be handled by the server.

[0263] In yet another example, there are instances in which a plurality of videos of approximately the same scene are captured by a plurality of terminals in, for example, a stadium, shopping mall, or factory. In such a case, for example, the encoding may be decentralized by dividing processing tasks between the plurality of terminals that captured the videos and, if necessary, other terminals that did not capture the videos, and the server, on a per-unit basis. The units may be, for example, groups of pictures (GOP), pictures, or tiles resulting from dividing a picture. This makes it possible to reduce load times and achieve streaming that is closer to real time.

[0264] Since the videos are of approximately the same scene, management and/or instructions may be carried out by the server so that the videos captured by the terminals can be cross-referenced. Moreover, the server may receive encoded data from the terminals, change the reference relationship between items of data, or correct or replace pictures themselves, and then perform the encoding. This makes it possible to generate a stream with increased quality and efficiency for the individual items of data.

[0265] Furthermore, the server may stream video data after performing transcoding to convert the encoding format of the video data. For example, the server may convert the encoding format from MPEG to VP (e.g., VP9), and may convert H.264 to H.265.

[0266] In this way, encoding can be performed by a terminal or one or more servers. Accordingly, although the device that performs the encoding is referred to as a "server" or "terminal" in the following description, some or all of the processes performed by the server may be performed by the terminal, and likewise some or all of the processes performed by the terminal may be performed by the server. This also applies to decoding processes.

[0267] (3D, Multi-angle)

There has been an increase in usage of images or videos combined from images or videos of different scenes concurrently captured, or of the same scene captured from different angles, by a plurality of terminals such as camera ex113 and/or smartphone ex115. Videos captured by the terminals are combined based on, for example, the separately obtained relative positional relationship between the terminals, or regions in a video having matching feature points.

[0268] In addition to the encoding of two-dimensional moving pictures, the server may encode a still image based on scene analysis of a moving picture, either automatically or at a point in time specified by the user, and transmit the encoded still image to a reception terminal. Furthermore, when the server can obtain the relative positional relationship between the video capturing terminals, in addition to two-dimensional moving pictures, the server can generate three-dimensional geometry of a scene based on video of the same scene captured from different angles. The server may separately encode three-dimensional data generated from, for example, a point cloud and, based on a result of recognizing or tracking a person or object using three-dimensional data, may select or reconstruct and generate a video to be transmitted to a reception terminal, from videos captured by a plurality of terminals.

[0269] This allows the user to enjoy a scene by freely selecting videos corresponding to the video capturing terminals, and allows the user to enjoy the content obtained by extracting a video at a selected viewpoint from three-dimensional data reconstructed from a plurality of images or videos. Furthermore, as with video, sound may be recorded from relatively different angles, and the server may multiplex audio from a specific angle or space with the corresponding video, and transmit the multiplexed video and audio.

[0270] In recent years, content that is a composite of the real world and a virtual world, such as virtual reality (VR) and augmented reality (AR) content, has also become popular. In the case of VR images, the server may create images from the viewpoints of both the left and right eyes, and perform encoding that tolerates reference between the two viewpoint images, such as multi-view coding (MVC), and, alternatively, may encode the images as separate streams without referencing. When the images are decoded as separate streams, the streams may be synchronized when reproduced, so as to recreate a virtual three-dimensional space in accordance with the viewpoint of the user.

[0271] In the case of AR images, the server superimposes virtual object information existing in a virtual space onto camera information representing a real-world space, based on a three-dimensional position or movement from the perspective of the user. The decoder may obtain or store virtual object information and three-dimensional data, generate two-dimensional images based on movement from the perspective of the user, and then generate superimposed data by seamlessly connecting the images. Alternatively, the decoder may transmit, to the server, motion from the perspective of the user in addition to a request for virtual object information. The server may generate superimposed data based on three-dimensional data stored in the server in accordance with the received motion, and encode and stream the generated superimposed data to the decoder. Note that superimposed data includes, in addition to RGB values, an α value indicating transparency, and the server sets the α value for sections other than the object generated from three-dimensional data to, for example, 0, and may perform the encoding while those sections are transparent. Alternatively, the server may set the background to a predetermined RGB value, such as a chroma key, and generate data in which areas other than the object are set as the background.

[0272] Decoding of similarly streamed data may be performed by the client (i.e., the terminals), on the server side, or divided therebetween. In one example, one terminal may transmit a reception request to a server, the requested content may be received and decoded by another terminal, and a decoded signal may be transmitted to a device having a display. It is possible to reproduce high image quality data by decentralizing processing and appropriately selecting content regardless of the processing ability of the communications terminal itself. In yet another example, while a TV, for example, is receiving image data that is large in size, a region of a picture, such as a tile obtained by dividing the picture, may be decoded and displayed on a personal terminal or terminals of a viewer or viewers of the TV. This makes it possible for the viewers to share a big-picture view as well as for each viewer to check his or her assigned area, or inspect a region in further detail up close.

[0273] In situations in which a plurality of wireless connections are possible over near, mid, and far distances, indoors or outdoors, it may be possible to seamlessly receive content using a streaming system standard such as MPEG-DASH. The user may switch between data in real time while freely selecting a decoder or display apparatus including the user's terminal, displays arranged indoors or outdoors, etc. Moreover, using, for example, information on the position of the user, decoding can be performed while switching which terminal handles decoding and which terminal handles the displaying of content. This makes it possible to map and display information, while the user is on the move in route to a destination, on the wall of a nearby building in which a device capable of displaying content is embedded, or on part of the ground.

Moreover, it is also possible to switch the bit rate of the received data based on the accessibility to the encoded data on a network, such as when encoded data is cached on a server quickly accessible from the reception terminal, or when encoded data is copied to an edge server in a content delivery service.

[0274] (Scalable Encoding)

The switching of content will be described with reference to a scalable stream, illustrated in FIG. 27, which is compression coded via implementation of the moving picture encoding method described in the above embodiments. The server may have a configuration in which content is switched while making use of the temporal and/or spatial scalability of a stream, which is achieved by division into and encoding of layers, as illustrated in FIG. 27. Note that there may be a plurality of individual streams that are of the same content but different quality. In other words, by determining which layer to decode based on internal factors, such as the processing ability on the decoder side, and external factors, such as communication bandwidth, the decoder side can freely switch between low resolution content and high resolution content while decoding. For example, in a case in which the user wants to continue watching, for example at home on a device such as a TV connected to the internet, a video that the user had been previously watching on smartphone ex115 while on the move, the device can simply decode the same stream up to a different layer, which reduces the server side load.

[0275] Furthermore, in addition to the configuration described above, in which scalability is achieved as a result of the pictures being encoded per layer, with the enhancement layer being above the base layer, the enhancement layer may include metadata based on, for example, statistical information on the image. The decoder side may generate high image quality content by performing super-resolution imaging on a picture in the base layer based on the metadata. Super-resolution imaging may improve the SN ratio while maintaining resolution and/or increasing resolution. Metadata includes information for identifying a linear or a non-linear filter coefficient, as used in super-resolution processing, or information identifying a parameter value in filter processing, machine learning, or a least squares method used in super-resolution processing.

[0276] Alternatively, a configuration may be provided in which a picture is divided into, for example, tiles in accordance with, for example, the meaning of an object in the image. On the decoder side, only a partial region is decoded by selecting a tile to decode. Further, by storing an attribute of the object (person, car, ball, etc.) and a position of the object in the video (coordinates in identical images) as metadata, the decoder side can identify the position of a desired object based on the metadata and determine which tile or tiles include that object. For example, as illustrated in FIG. 28, metadata may be stored using a data storage structure different from pixel data, such as an SEI (supplemental enhancement information) message in HEVC. This metadata indicates, for example, the position, size, or color of the main object.

[0277] Metadata may be stored in units of a plurality of pictures, such as stream, sequence, or random access units. The decoder side can obtain, for example, the time at which a specific person appears in the video, and by fitting the time information with picture unit information, can identify a picture in which the object is present, and can determine the position of the object in the picture.

[0278] (Web Page Optimization)

FIG. 29 illustrates an example of a display screen of a web page on computer ex111, for example. FIG. 30 illustrates an example of a display screen of a web page on smartphone ex115, for example. As illustrated in FIG. 29 and FIG. 30, a web page may include a plurality of image links that are links to image content, and the appearance of the web page differs depending on the device used to view the web page. When a plurality of image links are viewable on the screen, until the user explicitly selects an image link, or until the image link is in the approximate center of the screen or the entire image link fits in the screen, the display apparatus (decoder) may display, as the image links, still images included in the content or I pictures; may display video such as an animated gif using a plurality of still images or I pictures; or may receive only the base layer, and decode and display the video.

[0279] When an image link is selected by the user, the display apparatus performs decoding while giving the highest priority to the base layer. Note that if there is information in the HTML code of the web page indicating that the content is scalable, the display apparatus may decode up to the enhancement layer. Further, in order to guarantee real-time reproduction, before a selection is made or when the bandwidth is severely limited, the display apparatus can reduce delay between the point in time at which the leading picture is decoded and the point in time at which the decoded picture is displayed (that is, the delay between the start of the decoding of the content to the displaying of the content) by decoding and displaying only forward reference pictures (I picture, P picture, forward reference B picture). Still further, the display apparatus may purposely ignore the reference relationship between pictures, and coarsely decode all B and P pictures as forward reference pictures, and then perform normal decoding as the number of pictures received over time increases.

[0280] (Autonomous Driving)

When transmitting and receiving still image or video data such as two- or three-dimensional map information for autonomous driving or assisted driving of an automobile, the reception terminal may receive, in addition to image data belonging to one or more layers, information on, for example, the weather or road construction as metadata, and associate the metadata with the image data upon decoding. Note that metadata may be assigned per layer and, alternatively, may simply be multiplexed with the image data.

[0281] In such a case, since the automobile, drone, airplane, etc., containing the reception terminal is mobile, the reception terminal can seamlessly receive and perform decoding while switching between base stations among base stations ex106 through ex110 by transmitting information indicating the position of the reception terminal. Moreover, in accordance with the selection made by the user, the situation of the user, and/or the bandwidth of the connection, the reception terminal can dynamically select to what extent the metadata is received, or to what extent the map information, for example, is updated.

[0282] In content providing system ex100, the client can receive, decode, and reproduce, in real time, encoded information transmitted by the user.

[0283] (Streaming of Individual Content)

In content providing system ex100, in addition to high image quality, long content distributed by a video distribution entity, unicast or multicast streaming of low image quality, and short content from an individual are also possible. Such content from individuals is likely to further increase in popularity. The server may first perform editing processing on the content before the encoding processing, in order to refine the individual content. This may be achieved using the following configuration, for example.

[0284] In real time while capturing video or image content, or after the content has been captured and accumulated, the server performs recognition processing based on the raw data or encoded data, such as capture error processing, scene search processing, meaning analysis, and/or object detection processing. Then, based on the result of the recognition processing, the server - either when prompted or automatically - edits the content, examples of which include: correction such as focus and/or motion blur correction; removing low-priority scenes such as scenes that are low in brightness compared to other pictures, or out of focus; object edge adjustment; and color tone adjustment. The server encodes the edited data based on the result of the editing. It is known that excessively long videos tend to receive fewer views. Accordingly, in order to keep the content within a specific length that scales with the length of the original video, the server may, in addition to the low-priority scenes described above, automatically clip out scenes with low movement, based on an image processing result. Alternatively, the server may generate and encode a video digest based on a result of an analysis of the meaning of a scene.

[0285] There may be instances in which individual content may include content that infringes a copyright, moral right, portrait rights, etc. Such instance may lead to an unfavorable situation for the creator, such as when content is shared beyond the scope intended by the creator. Accordingly, before encoding, the server may, for example, edit images so as to blur faces of people in the periphery of the screen or blur the inside of a house, for example. Further, the server may be configured to recognize the faces of people other than a registered person in images to be encoded, and when such faces appear in an image, may apply a mosaic filter, for example, to the face of the person. Alternatively, as pre- or post-processing for encoding, the user may specify, for copyright reasons, a region of an image including a person or a region of the background to be processed. The server may process the specified region by, for example, replacing the region with a different image, or blurring the region. If the region includes a person, the person may be tracked in the moving picture, and the person's head region may be replaced with another image as the person moves.

[0286] Since there is a demand for real-time viewing of content produced by individuals, which tends to be small in data size, the decoder first receives the base layer as the highest priority, and performs decoding and reproduction, although this may differ depending on bandwidth. When the content is reproduced two or more times, such as when the decoder receives the enhancement layer during decoding and reproduction of the base layer, and loops the reproduction, the decoder may reproduce a high image quality video including the enhancement layer. If the stream is encoded using such scalable encoding, the video may be low quality when in an unselected state or at the start of the video, but it can offer an experience in which the image quality of the stream progressively increases in an intelligent manner. This is not limited to just scalable encoding; the same experience can be offered by configuring a single stream from a low quality stream reproduced for the first time and a second stream encoded using the first stream as a reference.

[0287] (Other Implementation and Application Examples)

The encoding and decoding may be performed by LSI (large scale integration circuitry) ex500 (see FIG. 26), which is typically included in each terminal. LSI ex500 may be configured of a single chip or a plurality of chips. Software for encoding and decoding moving pictures may be integrated into some type of a recording medium (such as a CD-ROM, a flexible disk, or a hard disk) that is readable by, for example, computer ex111, and the encoding and decoding may be performed using the software. Furthermore, when smartphone ex115 is equipped with a camera, the video data obtained by the camera may be transmitted. In this case, the video data is coded by LSI ex500 included in smartphone ex115.

[0288] Note that LSI ex500 may be configured to download and activate an application. In such a case, the terminal first determines whether it is compatible with the scheme used to encode the content, or whether it is capable of executing a specific service. When the terminal is not compatible with the encoding scheme of the content, or when the terminal is not capable of executing a specific service, the terminal first downloads a codec or application software and then obtains and reproduces the content.

[0289] Aside from the example of content providing system ex100 that uses internet ex101, at least the moving picture encoder (image encoder) or the moving picture decoder (image decoder) described in the above embodiments may be implemented in a digital broadcasting system. The same encoding processing and decoding processing may be applied to transmit and receive broadcast radio waves superimposed with multiplexed audio and video data using, for example, a satellite, even though this is geared toward multicast, whereas unicast is easier with content providing system ex100.

[0290] (Hardware Configuration)

FIG. 31 illustrates further details of smartphone ex115 shown in FIG. 26. FIG. 32 illustrates a configuration example of smartphone ex115. Smartphone ex115 includes antenna ex450 for transmitting and receiving radio waves to and from base station ex110, camera ex465 capable of capturing video and still images, and display ex458 that displays decoded data, such as video captured by camera ex465 and video received by antenna ex450. Smartphone ex115 further includes user interface ex466 such as a touch panel, audio output unit ex457 such as a speaker for outputting speech or other audio, audio input unit ex456 such as a microphone for audio input, memory ex467 capable of storing decoded data such as captured video or still images, recorded audio, received video or still images, and mail, as well as decoded data, and slot ex464 which is an interface for SIM ex468 for authorizing access to a network and various data. Note that external memory may be used instead of memory ex467.

[0291] Main controller ex460, which comprehensively controls display ex458 and user interface ex466, power supply circuit ex461, user interface input controller ex462, video signal processor ex455, camera interface ex463, display controller ex459, modulator/demodulator ex452, multiplexer/demultiplexer ex453, audio signal processor ex454, slot ex464, and memory ex467 are connected via bus ex470.

[0292] When the user turns on the power button of power supply circuit ex461, smartphone ex115 is powered on into an operable state, and each component is supplied with power from a battery pack.

[0293] Smartphone ex115 performs processing for, for example, calling and data transmission, based on control performed by main controller ex460, which includes a CPU, ROM, and RAM. When making calls, an audio signal recorded by audio input unit ex456 is converted into a digital audio signal by audio signal processor ex454, to which spread spectrum processing is applied by modulator/demodulator ex452 and digital-analog conversion, and frequency conversion processing is applied by transmitter/receiver ex451, and the resulting signal is transmitted via antenna ex450. The received data is amplified, frequency converted, and analog-digital converted, inverse spread spectrum processed by modulator/demodulator ex452, converted into an analog audio signal by audio signal processor ex454, and then output from audio output unit ex457. In data transmission mode, text, still-image, or video data is transmitted by main controller ex460 via user interface input controller ex462 based on operation of user interface ex466 of the main body, for example. Similar transmission and reception processing is performed. In data transmission mode, when sending a video, still image, or video and audio, video signal processor ex455 compression encodes, via the moving picture encoding method described in the above embodiments, a video signal stored in memory ex467 or a video signal input from camera ex465, and transmits the encoded video data to multiplexer/demultiplexer ex453. Audio signal processor ex454 encodes an audio signal recorded by audio input unit ex456 while camera ex465 is capturing a video or still image, and transmits the encoded audio data to multiplexer/demultiplexer ex453. Multiplexer/demultiplexer ex453 multiplexes the encoded video data and encoded audio data using a predetermined scheme, modulates and converts the data using modulator/demodulator (modulator/demodulator circuit) ex452 and transmitter/receiver ex451, and transmits the result via antenna ex450.

[0294] When video appended in an email or a chat, or a video linked from a web page, is received, for example, in order to decode the multiplexed data received via antenna ex450, multiplexer/demultiplexer ex453 demultiplexes the multiplexed data to divide the multiplexed data into a bitstream of video data and a bitstream of audio data, supplies the encoded video data to video signal processor ex455 via synchronous bus ex470, and supplies the encoded audio data to audio signal processor ex454 via synchronous bus ex470. Video signal processor ex455 decodes the video signal using a moving picture decoding method corresponding to the moving picture encoding method described in the above embodiments, and video or a still image included in the linked moving picture file is displayed on display ex458 via display controller ex459. Audio signal processor ex454 decodes the audio signal and outputs audio from audio output unit ex457. Since real-time streaming is becoming increasingly popular, there may be instances in which reproduction of the audio may be socially inappropriate, depending on the user's environment. Accordingly, as an initial value, a configuration in which only video data is reproduced, i.e., the audio signal is not reproduced, is preferable; audio may be synchronized and reproduced only when an input, such as when the user clicks video data, is received.

[0295] Although smartphone ex115 was used in the above example, three other implementations are conceivable: a transceiver terminal including both an encoder and a decoder; a transmitter terminal including only an encoder; and a receiver terminal including only a decoder. In the description of the digital broadcasting system, an example is given in which multiplexed data obtained as a result of video data being multiplexed with audio data is received or transmitted. The multiplexed data, however, may be video data multiplexed with data other than audio data, such as text data related to the video. Further, the video data itself rather than multiplexed data may be received or transmitted.

[0296] Although main controller ex460 including a CPU is described as controlling the encoding or decoding processes, various terminals often include GPUs. Accordingly, a configuration is acceptable in which a large area is processed at once by making use of the performance ability of the GPU via memory shared by the CPU and GPU, or memory including an address that is managed so as to allow common usage by the CPU and GPU. This makes it possible to shorten encoding time, maintain the real-time nature of the stream, and reduce delay. In particular, processing relating to motion estimation, deblocking filtering, sample adaptive offset (SAO), and transformation/quantization can be effectively carried out by the GPU instead of the CPU in units of pictures, for example, all at once.

Claims

[Claim 1] An image encoder comprising:
circuitry; and
a memory coupled to the circuitry;
wherein the circuitry, in operation, performs:
splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition;
predicting a first motion vector for the first partition and a second motion vector for the second partition; and
encoding the first partition using the first motion vector and the second partition using the second motion vector.

[Claim 2] The encoder of claim 1, wherein the second partition has a non-rectangular shape.

[Claim 3] The encoder of claim 1, wherein the non-rectangular shape is a triangle.

[Claim 4] The encoder of claim 1, wherein the non-rectangular shape is selected from a group consisting of a triangle, a trapezoid, and a polygon with at least five sides and angles.

[Claim 5] The encoder of claim 1, wherein the predicting includes selecting the first motion vector from a first set of motion vector candidates and selecting the second motion vector from a second set of motion vector candidates.

[Claim 6] The encoder of claim 5, wherein the first set of motion vector candidates includes motion vectors of partitions neighboring the first partition, and the second set of motion vector candidates includes motion vectors of partitions neighboring the second partition.

[Claim 7] The encoder of claim 6, wherein the partitions neighboring the first partition and the partitions neighboring the second partition are outside of the image block from which the first partition and the second partition are split.

[Claim 8] The encoder of claim 6, wherein the neighboring partitions are one or both of spatially neighboring partitions and temporary neighboring partitions.

[Claim 9] The encoder of claim 5, wherein the first set of motion vector candidates is the same as the second set of motion vector candidates.

[Claim 10] The encoder of claim 5, wherein at least one of the first set of motion vector candidates and the second set of motion vector candidates is the same as a third set of motion vector candidates for the image block.

[Claim 11] The encoder of claim 5, wherein the circuitry,
in response to determining that the second partition has a non-rectangular shape, creating the second set of motion vector candidates that includes motion vectors of partitions neighboring the second partition exclusive of the first partition; and
in response to determining that the second partition has a rectangular shape, creating the second set of motion vector candidates that includes motion vectors of partitions neighboring the second partition inclusive of the first partition.

[Claim 12] The encoder of claim 1, wherein the predicting includes,
selecting a first motion vector candidate from a first set of motion vector candidates and deriving the first motion vector by adding a first motion vector difference to the first motion vector candidate, and
selecting a second motion vector candidate from a second set of motion vector candidates and deriving the second motion vector by adding a second motion vector difference to the second motion vector candidate.

[Claim 13] An image encoder comprising:
a splitter which, in operation, receives and splits an original picture into blocks,
an adder which, in operation, receives the blocks from the splitter and predictions from a prediction controller, and subtracts each prediction from its corresponding block to output a residual,
a transformer which, in operation, performs a transform on the residuals outputted from the adder to output transform coefficients,
a quantizer which, in operation, quantizes the transform coefficients to generate quantized transform coefficients,
an entropy encoder which, in operation, encodes the quantized transform coefficients to generate a bitstream, and
the prediction controller coupled to an inter predictor, an intra predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in an encoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an encoded reference block in a current picture,
wherein,
the prediction controller, in operation,
splits the blocks into a plurality of partitions including a first partition having a non-rectangular shape and a second partition,

predicts a first motion vector for the first partition and a second motion vector for the second partition, and
encodes the first partition using the first motion vector and the second partition using the second motion vector.

[Claim 14] The encoder of claim 13, wherein the second partition has a non-rectangular shape.

[Claim 15] The encoder of claim 13, wherein the non-rectangular shape is a triangle.

[Claim 16] An image encoding method comprising:
splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition;
predicting a first motion vector for the first partition and a second motion vector for the second partition; and
encoding the first partition using the first motion vector and the second partition using the second motion vector.

[Claim 17] The method of claim 16, wherein the second partition has a non-rectangular shape.

[Claim 18] The method of claim 16, wherein the non-rectangular shape is a triangle.

[Claim 19] An image decoder comprising:
circuitry;
a memory coupled to the circuitry;
wherein the circuitry, in operation, performs:
splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition;
predicting a first motion vector for the first partition and a second motion vector for the second partition; and
decoding the first partition using the first motion vector and the second partition using the second motion vector.

[Claim 20] The decoder of claim 19, wherein the second partition has a non-rectangular shape.

[Claim 21] The decoder of claim 19, wherein the non-rectangular shape is a triangle.

[Claim 22] The decoder of claim 19, wherein the non-rectangular shape is selected from a group consisting of a triangle, a trapezoid, and a polygon with at least five sides and angles.

[Claim 23] The decoder of claim 19, wherein the predicting includes selecting the first motion vector from a first set of motion vector candidates and selecting the second motion vector from a second set of motion vector candidates.

[Claim 24] The decoder of claim 23, wherein the first set of motion vector candidates includes motion vectors of partitions neighboring the first partition, and the second set of motion vector candidates includes motion vectors of partitions neighboring the second partition.

[Claim 25] The decoder of claim 24, wherein the partitions neighboring the first partition and the partitions neighboring the second partition are outside of the image block from which the first partition and the second partition are split.

[Claim 26] The decoder of claim 24, wherein the neighboring partitions are one or both of spatially neighboring partitions and temporary neighboring partitions.

[Claim 27] The decoder of claim 23, wherein the first set of motion vector candidates is the same as the second set of motion vector candidates.

[Claim 28] The decoder of claim 23, wherein at least one of the first set of motion vector candidates and the second set of motion vector candidates is the same as a third set of motion vector candidates for the image block.

[Claim 29] The decoder of claim 23, wherein the circuitry,
in response to determining that the second partition has a non-rectangular shape, creating the second set of motion vector candidates that includes motion vectors of partitions neighboring the second partition exclusive of the first partition; and
in response to determining that the second partition has a rectangular shape, creating the second set of motion vector candidates that includes motion vectors of partitions neighboring the second partition inclusive of the first partition.

[Claim 30] An image decoder comprising:
an entropy decoder which, in operation, receives and decodes an encoded bitstream to obtain quantized transform coefficients,
an inverse quantizer and transformer which, in operation, inverse quantizes the quantized transform coefficients to obtain transform coefficients and inverse transform the transform coefficients to obtain residuals,
an adder which, in operation, adds the residuals outputted from the inverse quantizer and transformer and predictions outputted from a prediction controller to reconstruct blocks, and
the prediction controller coupled to an inter predictor, an intra

predictor, and a memory, wherein the inter predictor, in operation, generates a prediction of a current block based on a reference block in a decoded reference picture and the intra predictor, in operation, generates a prediction of a current block based on an decoded reference block in a current picture,
wherein.
the prediction controller, in operation, splits an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition; predicts a first motion vector for the first partition and a second motion vector for the second partition; and decodes the first partition using the first motion vector and the second partition using the second motion vector.

[Claim 31] The decoder of claim 30, wherein the second partition has a non-rectangular shape.

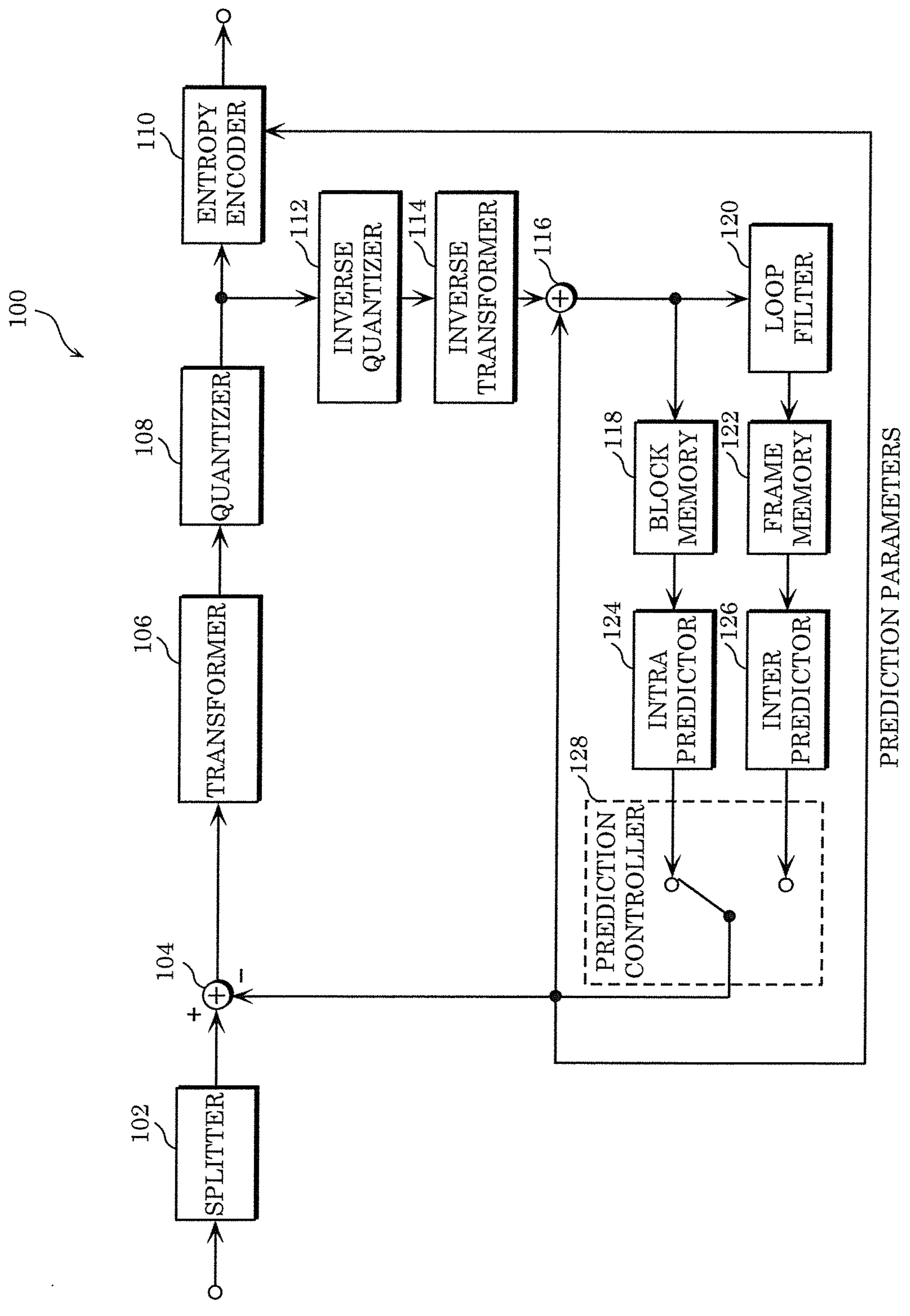
[Claim 32] The decoder of claim 30, wherein the non-rectangular shape is a triangle.

[Claim 33] An image decoding method comprising:
splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition;
predicting a first motion vector for the first partition and a second motion vector for the second partition; and
decoding the first partition using the first motion vector and the second partition using the second motion vector.

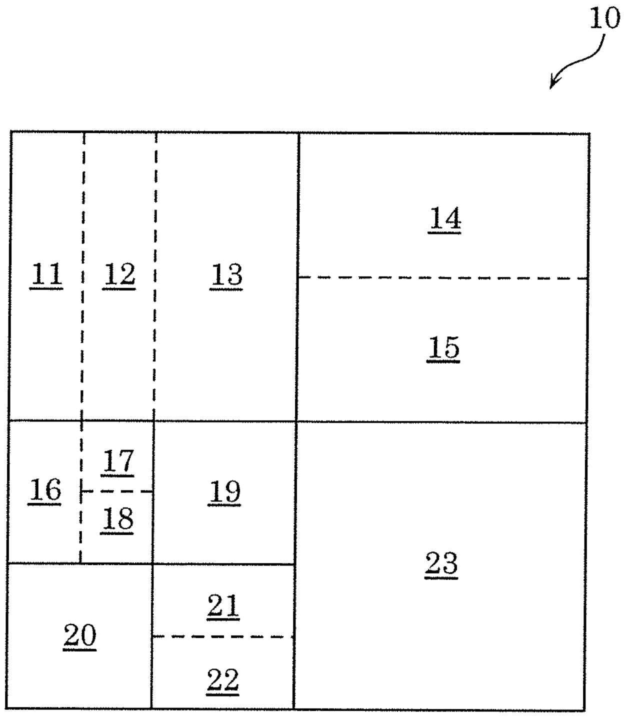
[Claim 34] The method of claim 33 wherein the second partition has a non-rectangular shape.

[Claim 35] The method of claim 33, wherein the non-rectangular shape is a triangle.

[Fig. 1]


100
102
SPLITTER
104
+
-
106
TRANSFORMER
108
QUANTIZER
110
ENTROPY ENCODER
112
INVERSE QUANTIZER
114
INVERSE TRANSFORMER
116
118
BLOCK MEMORY
120
LOOP FILTER
122
FRAME MEMORY
124
INTRA PREDICTOR
126
INTER PREDICTOR
128
PREDICTION CONTROLLER
PREDICTION PARAMETERS

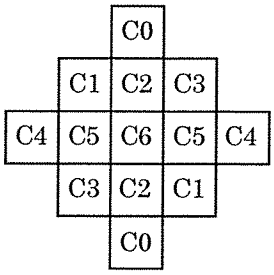
[Fig. 2]



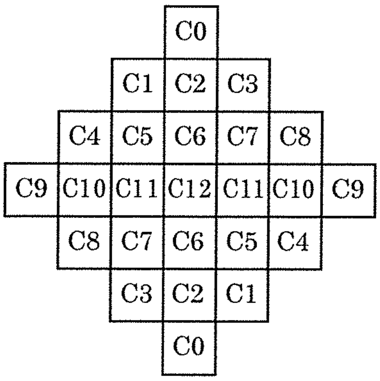
[Fig. 3]

TRANSFORM TYPE	BASIS FUNCTION $T_i(j)$, $i, j = 0, 1, ..., N - 1$
DCT - II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ WHERE $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT - V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right)$ WHERE $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$, $\omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
DCT - VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST - I	$T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$
DST - VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

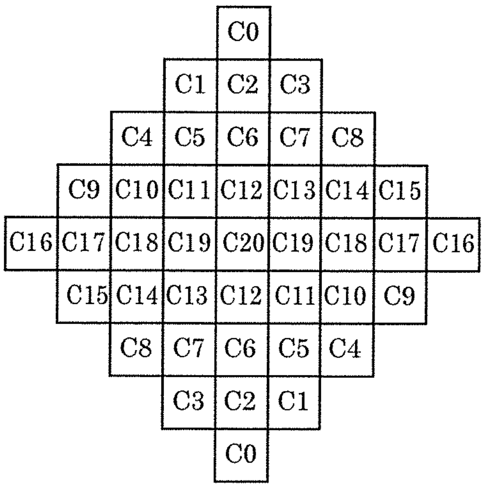
[Fig. 4A]



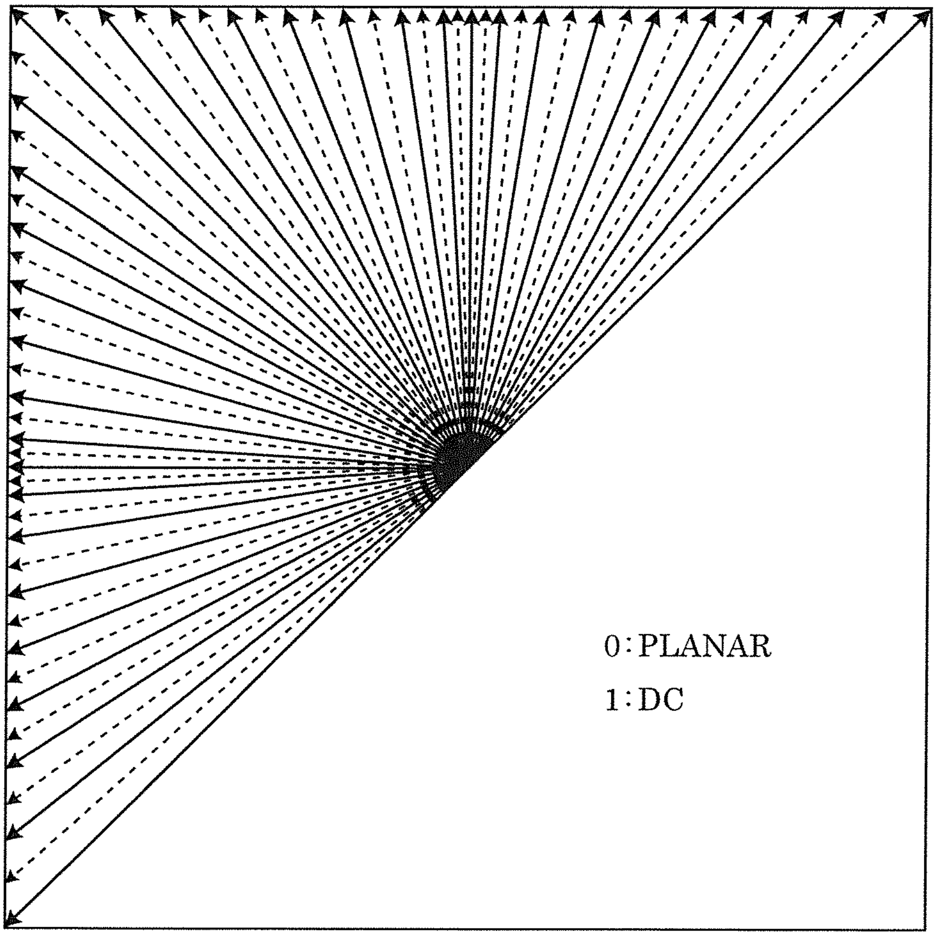
[Fig. 4B]



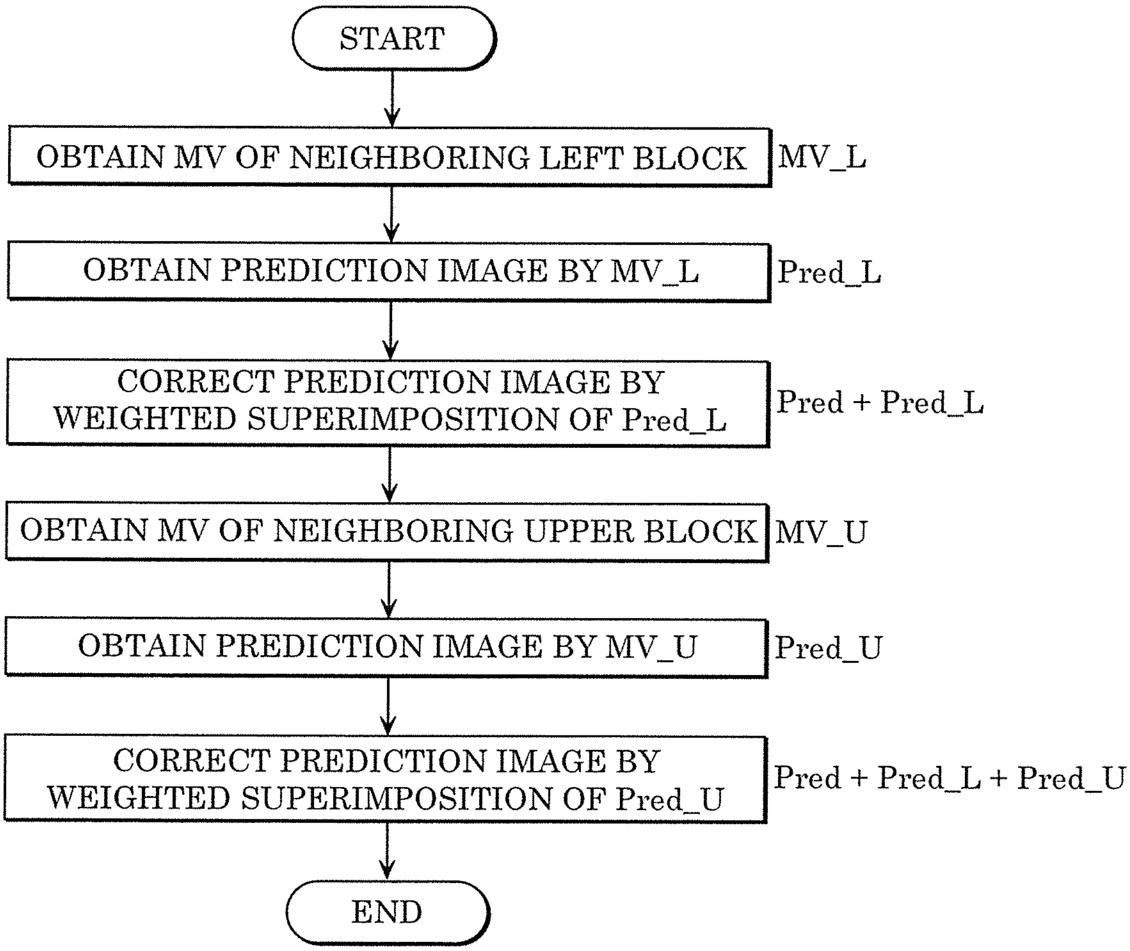
[Fig. 4C]



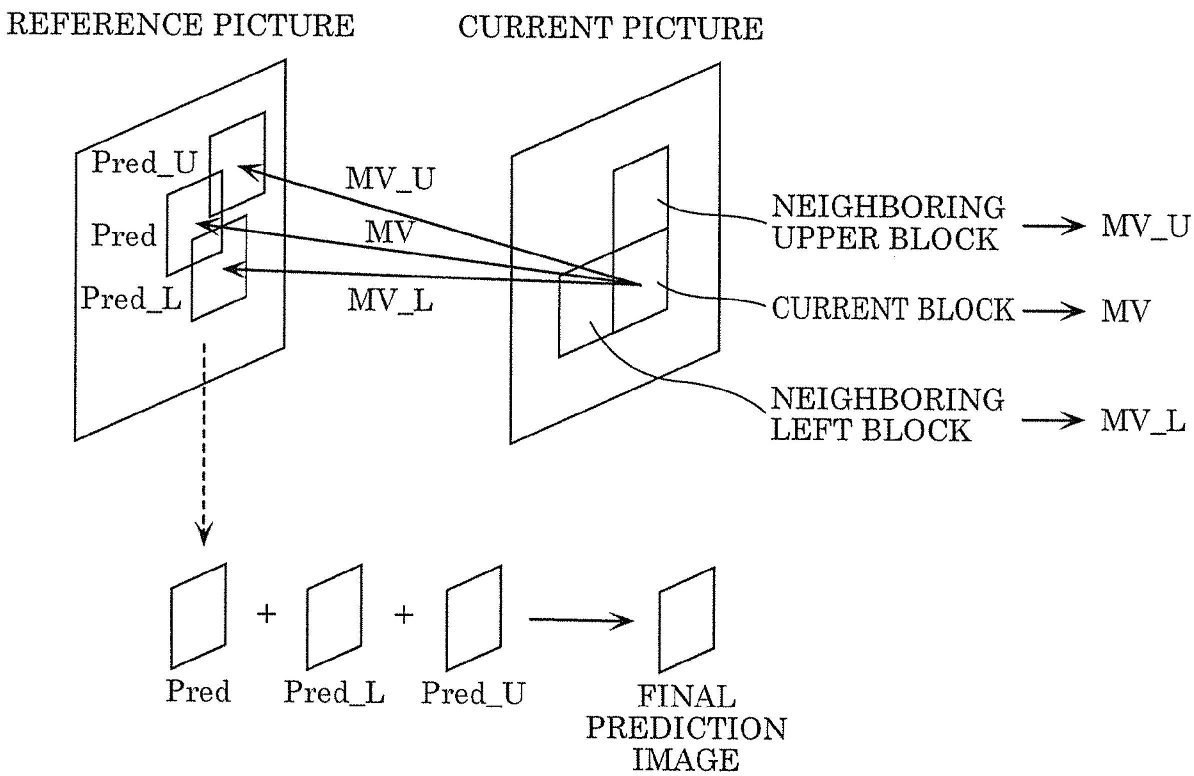
[Fig. 5A]



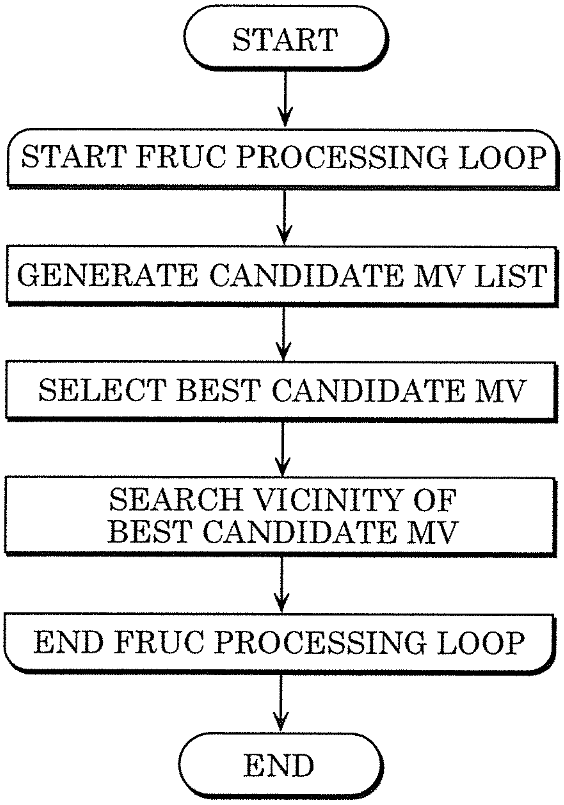
[Fig. 5B]


START
OBTAIN MV OF NEIGHBORING LEFT BLOCK
MV_L
OBTAIN PREDICTION IMAGE BY MV_L
Pred_L
CORRECT PREDICTION IMAGE BY WEIGHTED SUPERIMPOSITION OF Pred_L
Pred + Pred_L
OBTAIN MV OF NEIGHBORING UPPER BLOCK
MV_U
OBTAIN PREDICTION IMAGE BY MV_U
Pred_U
CORRECT PREDICTION IMAGE BY WEIGHTED SUPERIMPOSITION OF Pred_U
Pred + Pred_L + Pred_U
END

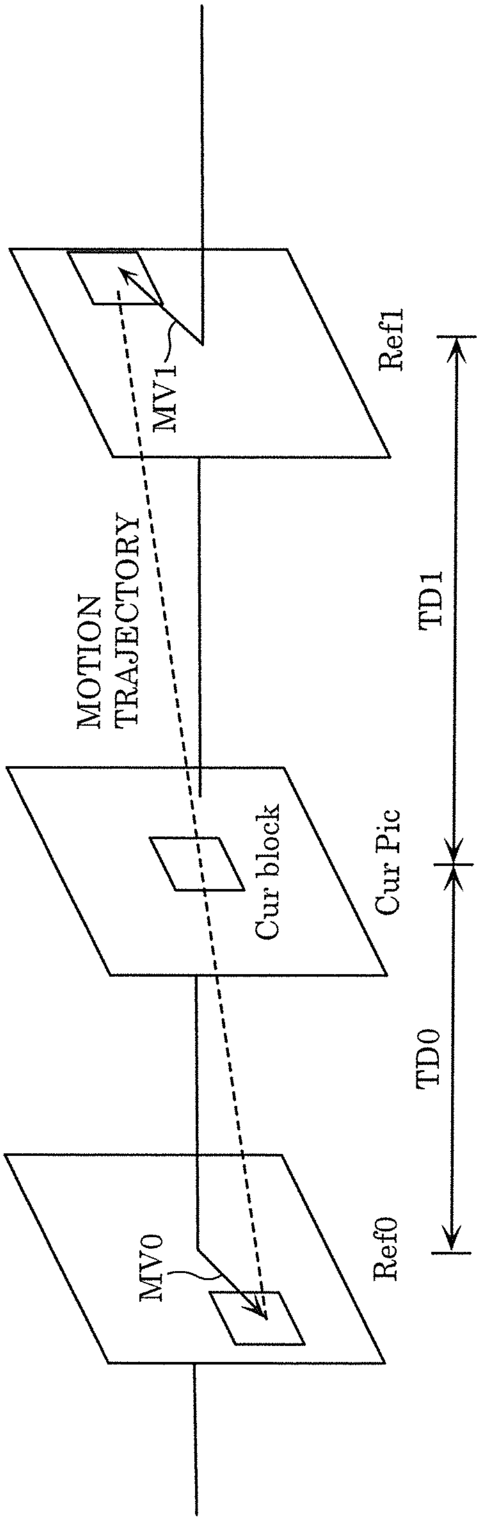
[Fig. 5C]


REFERENCE PICTURE
CURRENT PICTURE
Pred_U
Pred
Pred_L
MV_U
MV
MV_L
NEIGHBORING UPPER BLOCK
MV_U
CURRENT BLOCK
MV
NEIGHBORING LEFT BLOCK
MV_L
Pred
+
Pred_L
+
Pred_U
FINAL PREDICTION IMAGE

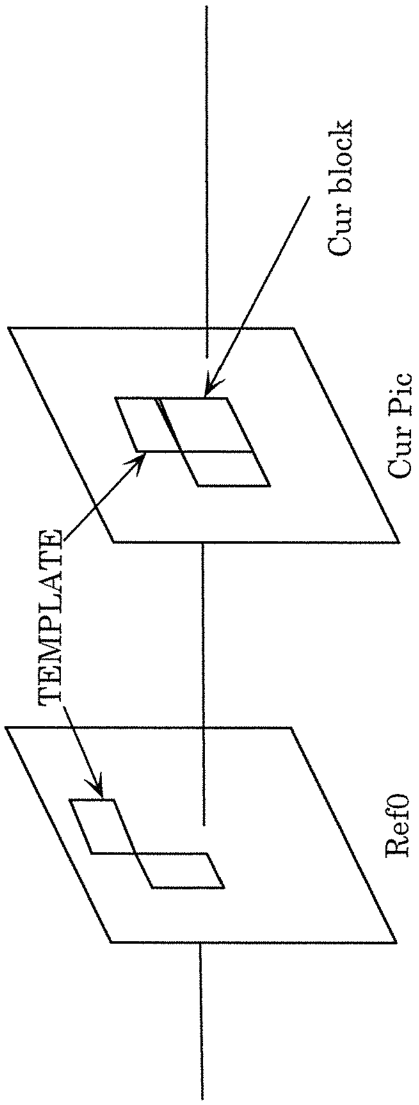
[Fig. 5D]


START
START FRUC PROCESSING LOOP
GENERATE CANDIDATE MV LIST
SELECT BEST CANDIDATE MV
SEARCH VICINITY OF BEST CANDIDATE MV
END FRUC PROCESSING LOOP
END

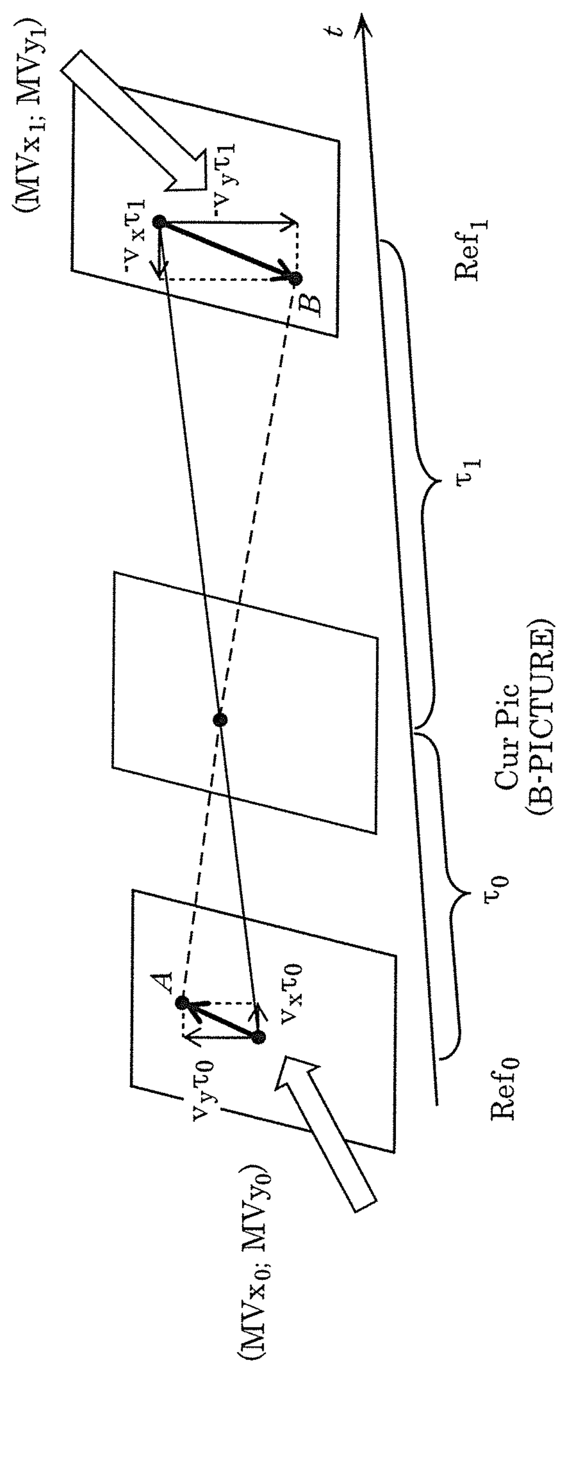
[Fig. 6]


MV1
Ref1
MOTION
TRAJECTORY
TD1
Cur block
Cur Pic
TD0
MV0
Ref0

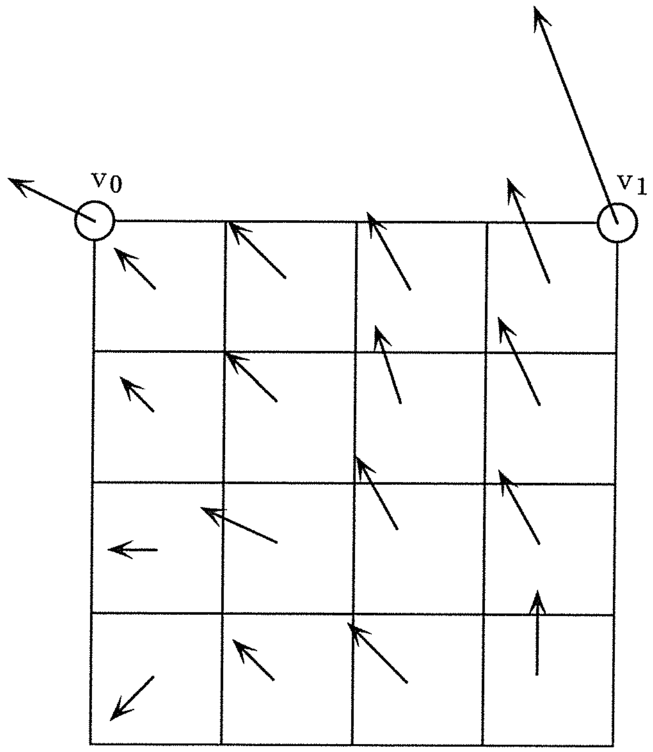
[Fig. 7]



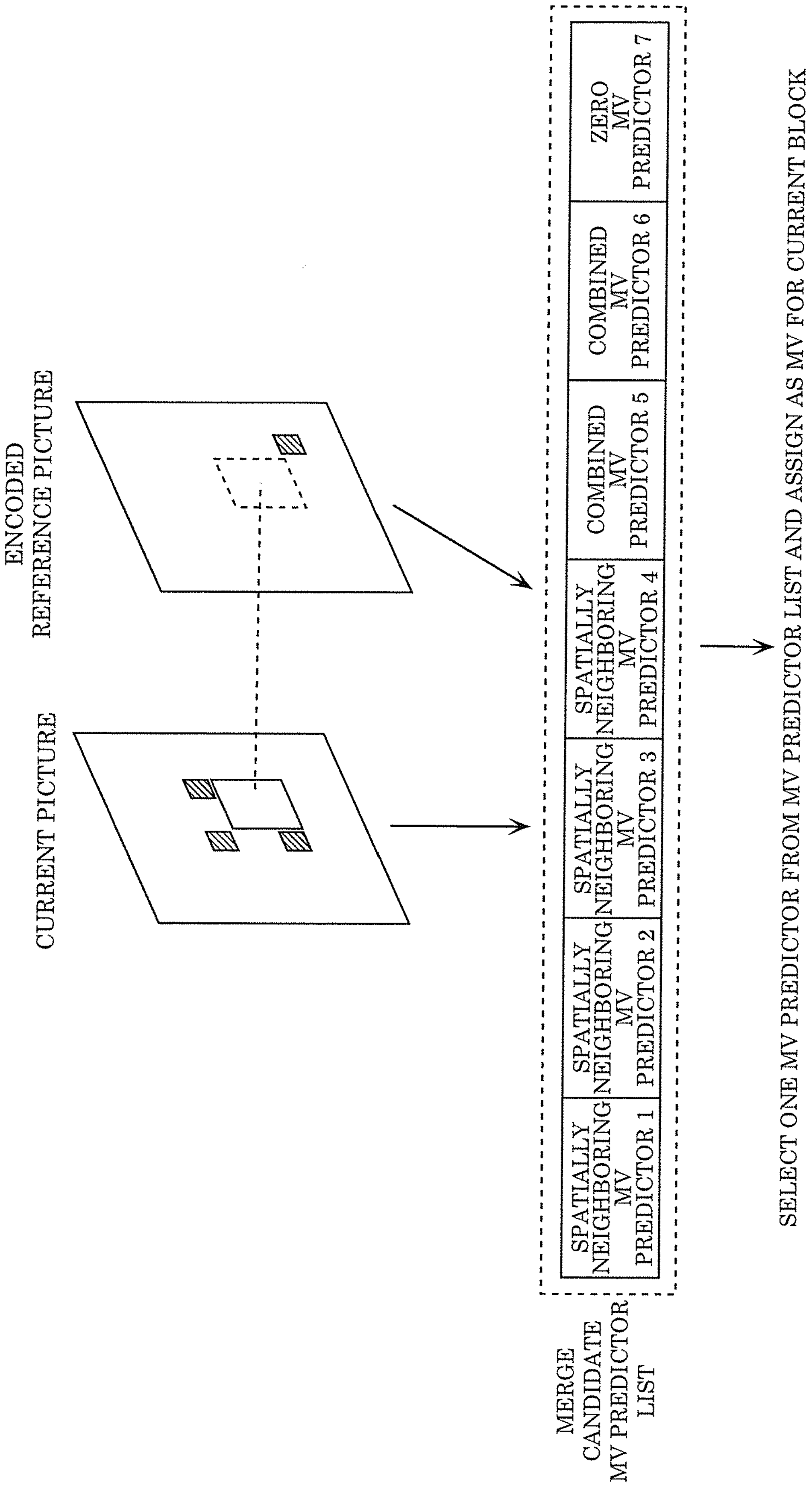
[Fig. 8]


$(MVx_1; MVy_1)$
$-v_y\tau_1$
$-v_x\tau_1$
B
t
Ref_1
τ_1
Cur Pic
(B-PICTURE)
τ_0
A
$v_x\tau_0$
$v_y\tau_0$
Ref_0
$(MVx_0; MVy_0)$

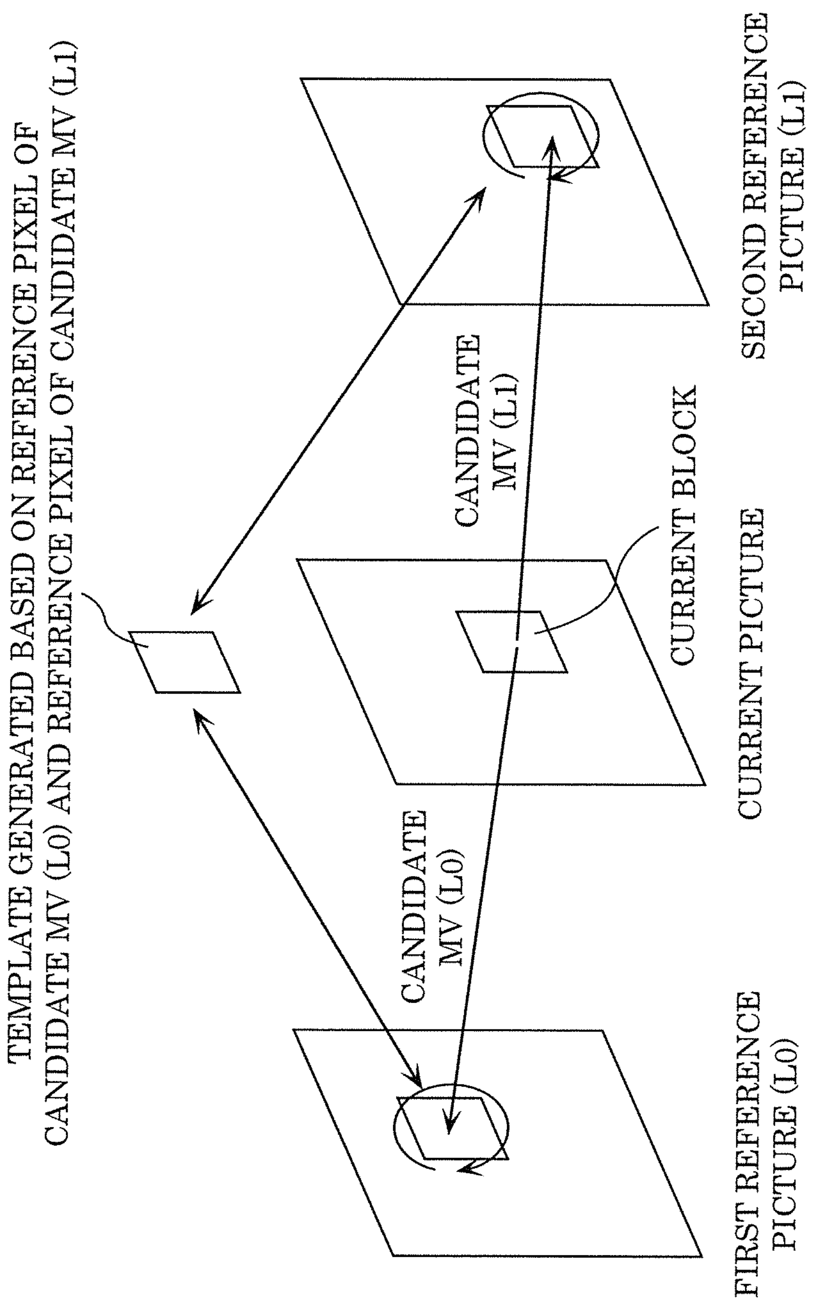
[Fig. 9A]



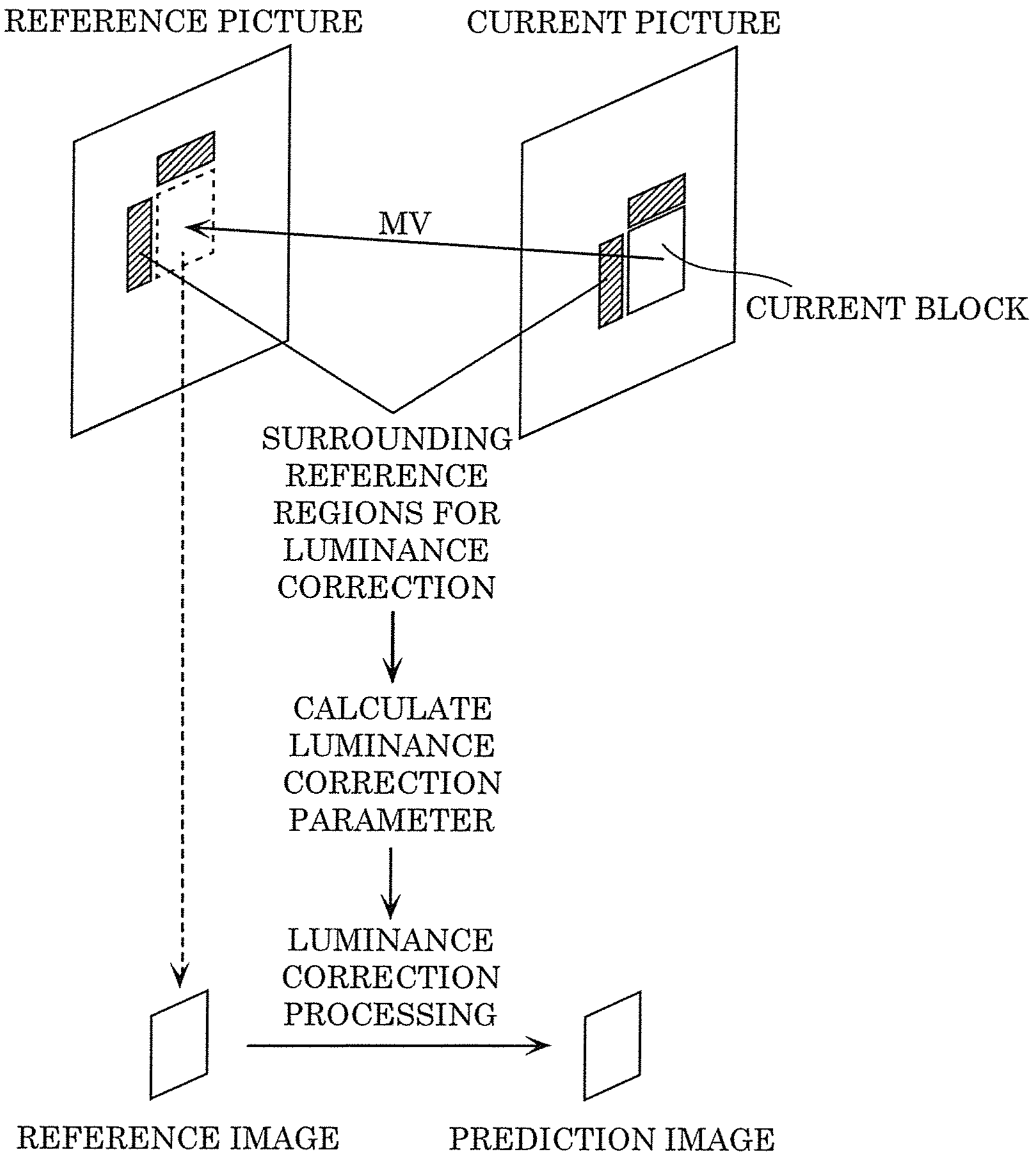
[Fig. 9B]


CURRENT PICTURE
ENCODED REFERENCE PICTURE
SPATIALLY NEIGHBORING MV PREDICTOR 1
SPATIALLY NEIGHBORING MV PREDICTOR 2
SPATIALLY NEIGHBORING MV PREDICTOR 3
SPATIALLY NEIGHBORING MV PREDICTOR 4
COMBINED MV PREDICTOR 5
COMBINED MV PREDICTOR 6
ZERO MV PREDICTOR 7
MERGE CANDIDATE MV PREDICTOR LIST
SELECT ONE MV PREDICTOR FROM MV PREDICTOR LIST AND ASSIGN AS MV FOR CURRENT BLOCK

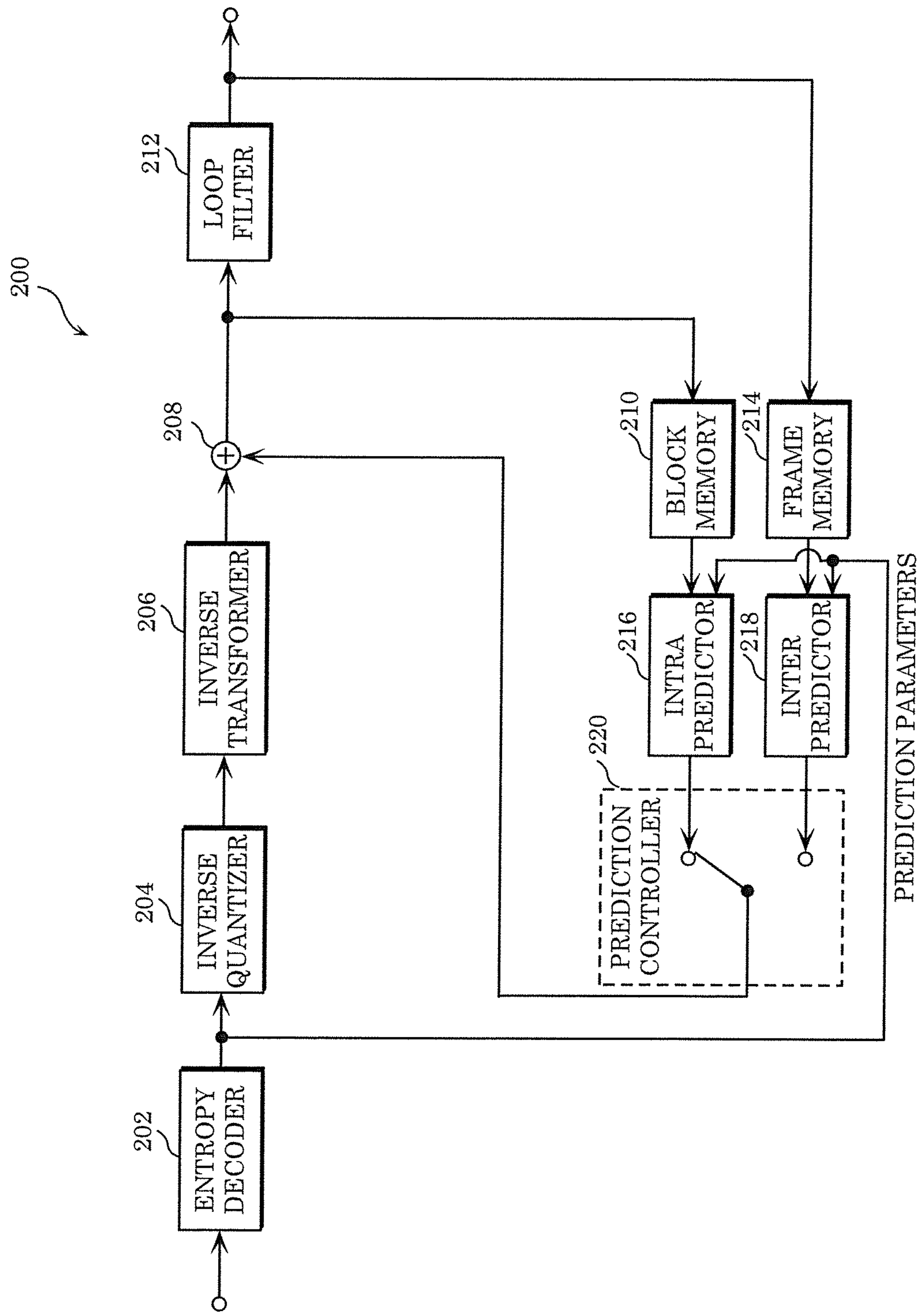
[Fig. 9C]


TEMPLATE GENERATED BASED ON REFERENCE PIXEL OF CANDIDATE MV (L0) AND REFERENCE PIXEL OF CANDIDATE MV (L1)
CANDIDATE MV (L1)
CANDIDATE MV (L0)
CURRENT BLOCK
FIRST REFERENCE PICTURE (L0)
CURRENT PICTURE
SECOND REFERENCE PICTURE (L1)

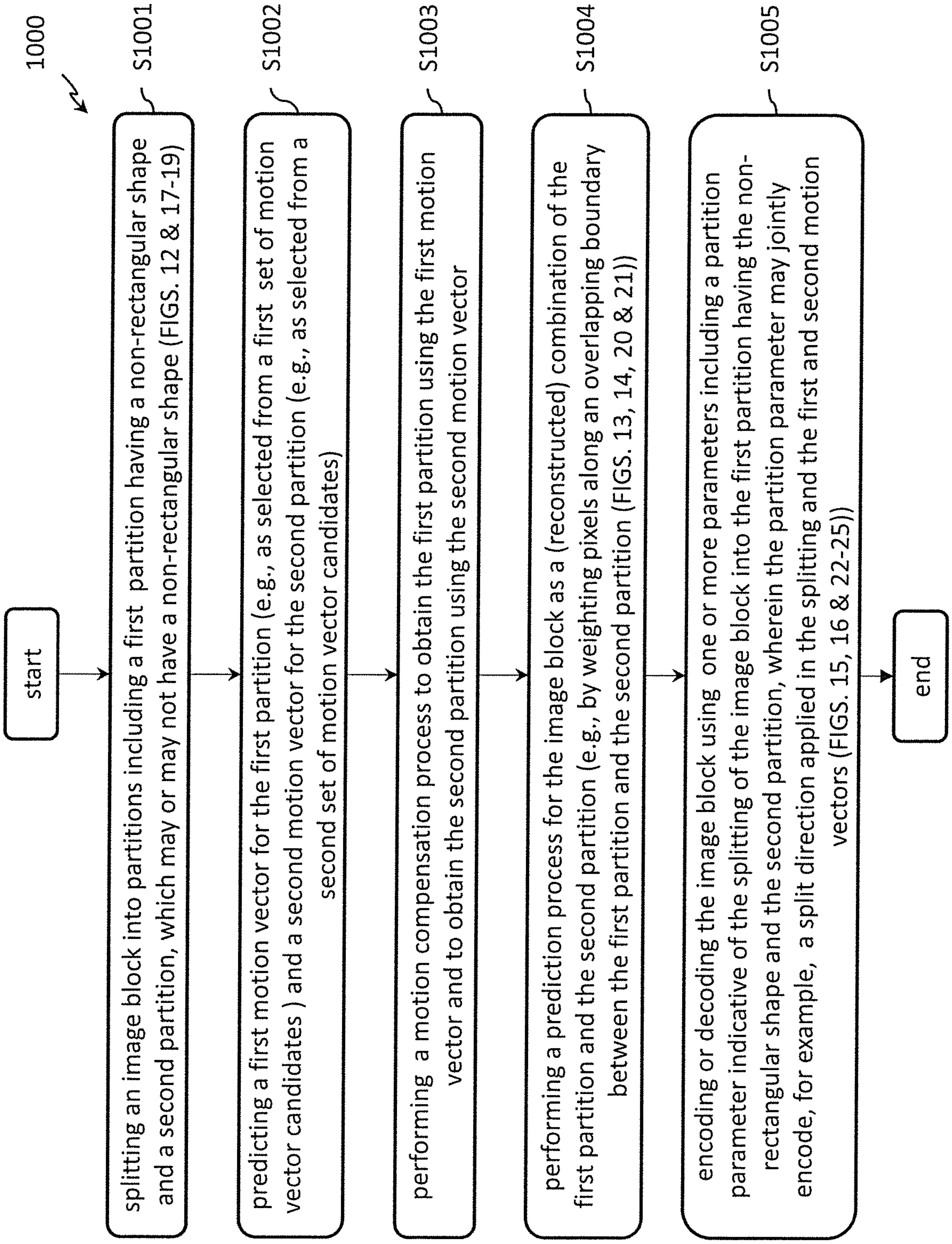
[Fig. 9D]


REFERENCE PICTURE
CURRENT PICTURE
MV
CURRENT BLOCK
SURROUNDING REFERENCE REGIONS FOR LUMINANCE CORRECTION
CALCULATE LUMINANCE CORRECTION PARAMETER
LUMINANCE CORRECTION PROCESSING
REFERENCE IMAGE
PREDICTION IMAGE

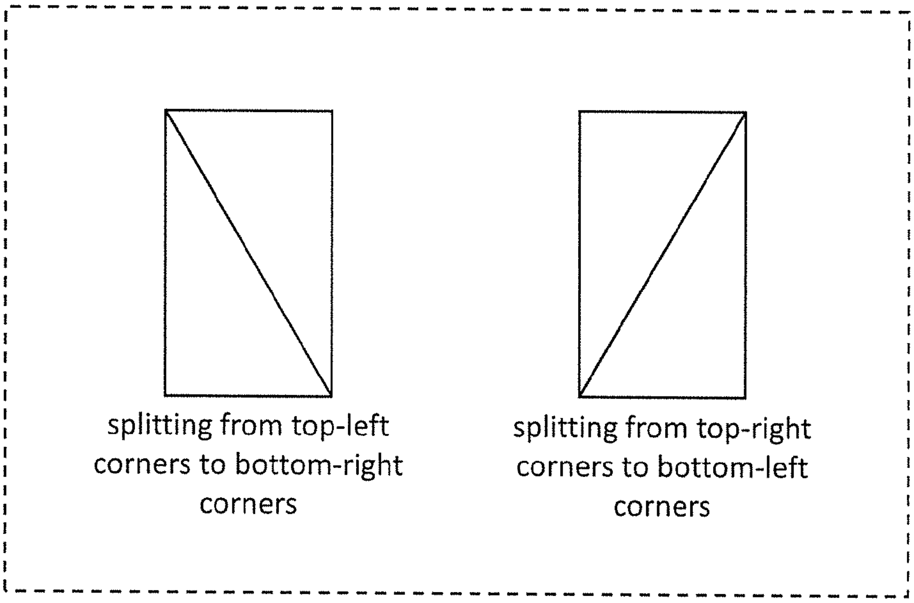
[Fig. 10]


200
202
ENTROPY DECODER
204
INVERSE QUANTIZER
206
INVERSE TRANSFORMER
208
212
LOOP FILTER
210
BLOCK MEMORY
214
FRAME MEMORY
216
INTRA PREDICTOR
218
INTER PREDICTOR
220
PREDICTION CONTROLLER
PREDICTION PARAMETERS

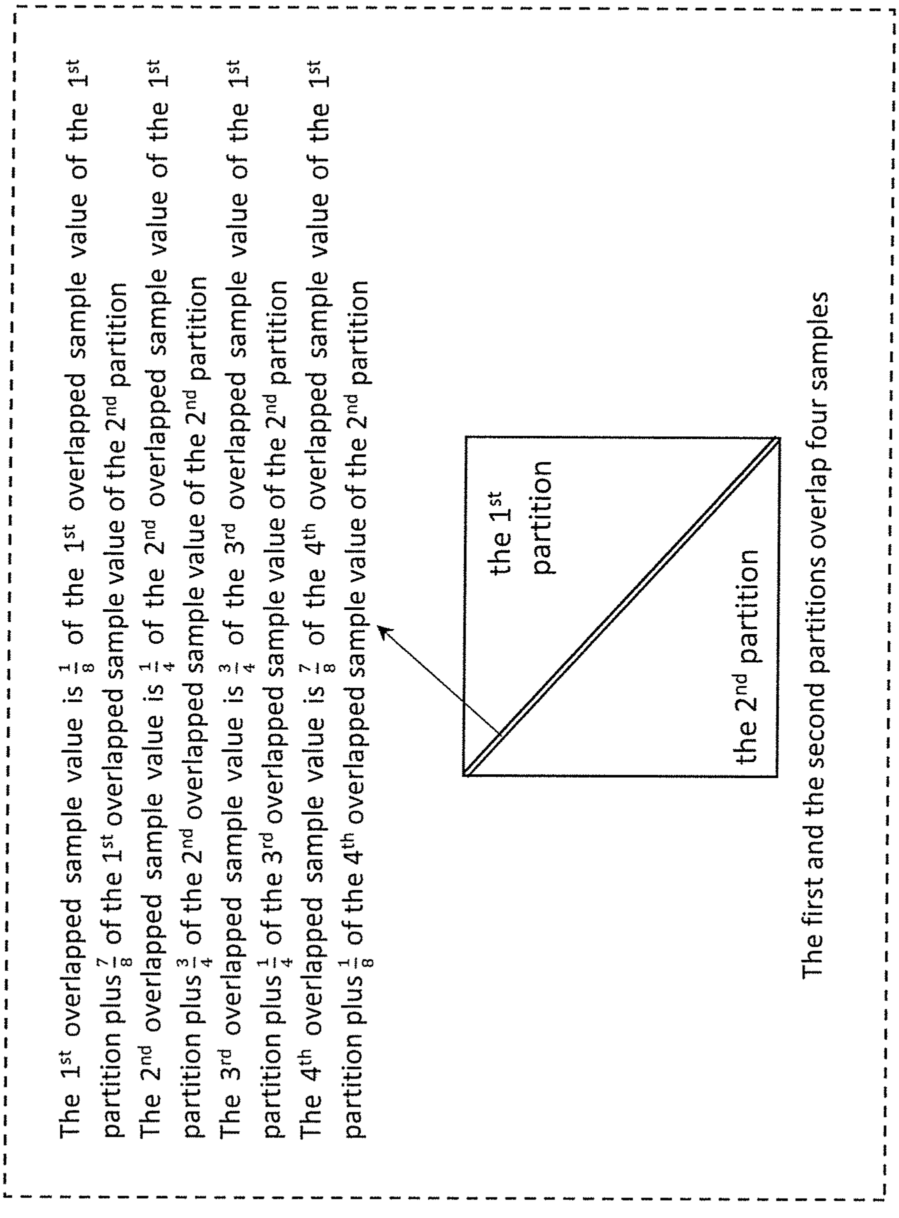
[Fig. 11]


1000
start
S1001
splitting an image block into partitions including a first partition having a non-rectangular shape and a second partition, which may or may not have a non-rectangular shape (FIGS. 12 & 17-19)
S1002
predicting a first motion vector for the first partition (e.g., as selected from a first set of motion vector candidates) and a second motion vector for the second partition (e.g., as selected from a second set of motion vector candidates)
S1003
performing a motion compensation process to obtain the first partition using the first motion vector and to obtain the second partition using the second motion vector
S1004
performing a prediction process for the image block as a (reconstructed) combination of the first partition and the second partition (e.g., by weighting pixels along an overlapping boundary between the first partition and the second partition (FIGS. 13, 14, 20 & 21))
S1005
encoding or decoding the image block using one or more parameters including a partition parameter indicative of the splitting of the image block into the first partition having the non-rectangular shape and the second partition, wherein the partition parameter may jointly encode, for example, a split direction applied in the splitting and the first and second motion vectors (FIGS. 15, 16 & 22-25))
end

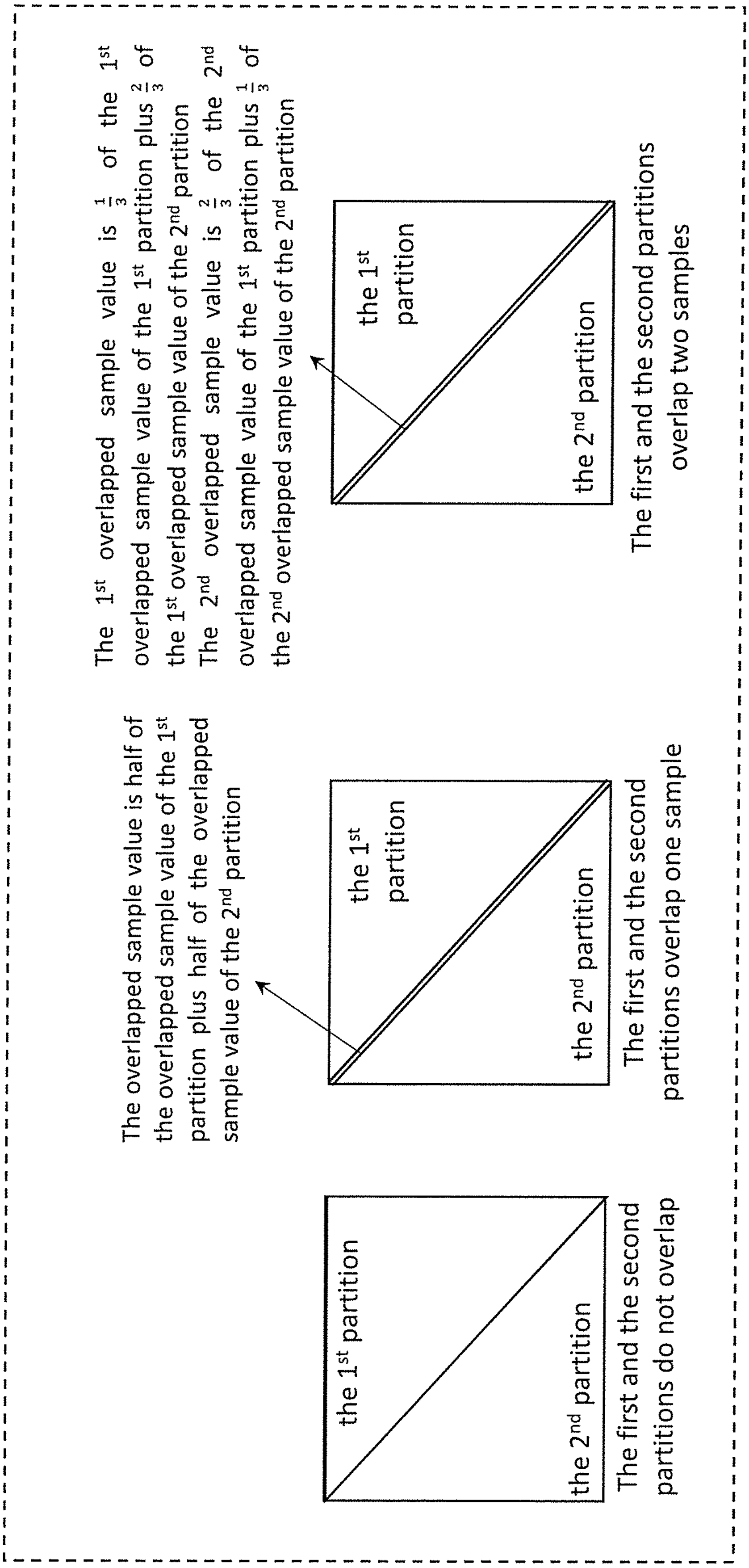
[Fig. 12]


splitting from top-left corners to bottom-right corners
splitting from top-right corners to bottom-left corners

[Fig. 13]


The 1st overlapped sample value is $\frac{1}{8}$ of the 1st overlapped sample value of the 1st partition plus $\frac{7}{8}$ of the 1st overlapped sample value of the 2nd partition
The 2nd overlapped sample value is $\frac{1}{4}$ of the 2nd overlapped sample value of the 1st partition plus $\frac{3}{4}$ of the 2nd overlapped sample value of the 2nd partition
The 3rd overlapped sample value is $\frac{3}{4}$ of the 3rd overlapped sample value of the 1st partition plus $\frac{1}{4}$ of the 3rd overlapped sample value of the 2nd partition
The 4th overlapped sample value is $\frac{7}{8}$ of the 4th overlapped sample value of the 1st partition plus $\frac{1}{8}$ of the 4th overlapped sample value of the 2nd partition
the 1st partition
the 2nd partition
The first and the second partitions overlap four samples

[Fig. 14]


the 1st partition
the 2nd partition
The first and the second partitions do not overlap
The overlapped sample value is half of the overlapped sample value of the 1st partition plus half of the overlapped sample value of the 2nd partition
the 1st partition
the 2nd partition
The first and the second partitions overlap one sample
The 1st overlapped sample value is $\frac{1}{3}$ of the 1st overlapped sample value of the 1st partition plus $\frac{2}{3}$ of the 1st overlapped sample value of the 2nd partition
The 2nd overlapped sample value is $\frac{2}{3}$ of the 2nd overlapped sample value of the 1st partition plus $\frac{1}{3}$ of the 2nd overlapped sample value of the 2nd partition
the 1st partition
the 2nd partition
The first and the second partitions overlap two samples

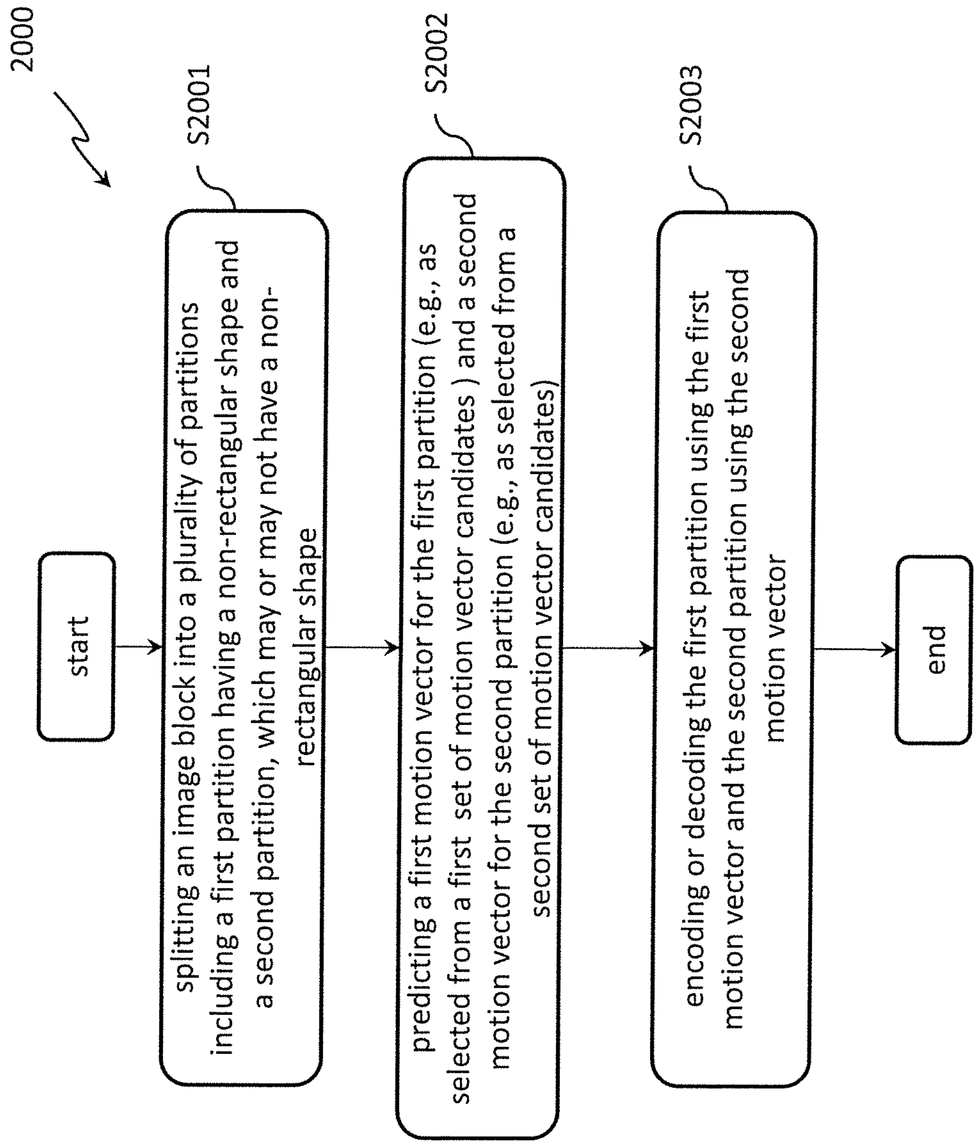
[Fig. 15]

the first index value	the direction of splitting the image block into two triangles	the motion vector of the first partition	the motion vector of the second partition
0	From top-left corner to bottom-right corner	the second motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
1	From top-right corner to bottom-left corner	the first motion vector in the first motion vector candidate prediction list	the second motion vector in the second motion vector candidate prediction list
2	From top-right corner to bottom-left corner	the second motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
3	From top-left corner to bottom-right corner	the second motion vector in the first motion vector candidate prediction list	the second motion vector in the second motion vector candidate prediction list
4	From top-right corner to bottom-left corner	the second motion vector in the first motion vector candidate prediction list	the third motion vector in the second motion vector candidate prediction list
5	From top-left corner to bottom-right corner	the third motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
6	From top-left corner to bottom-right corner	the fourth motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
⋮			

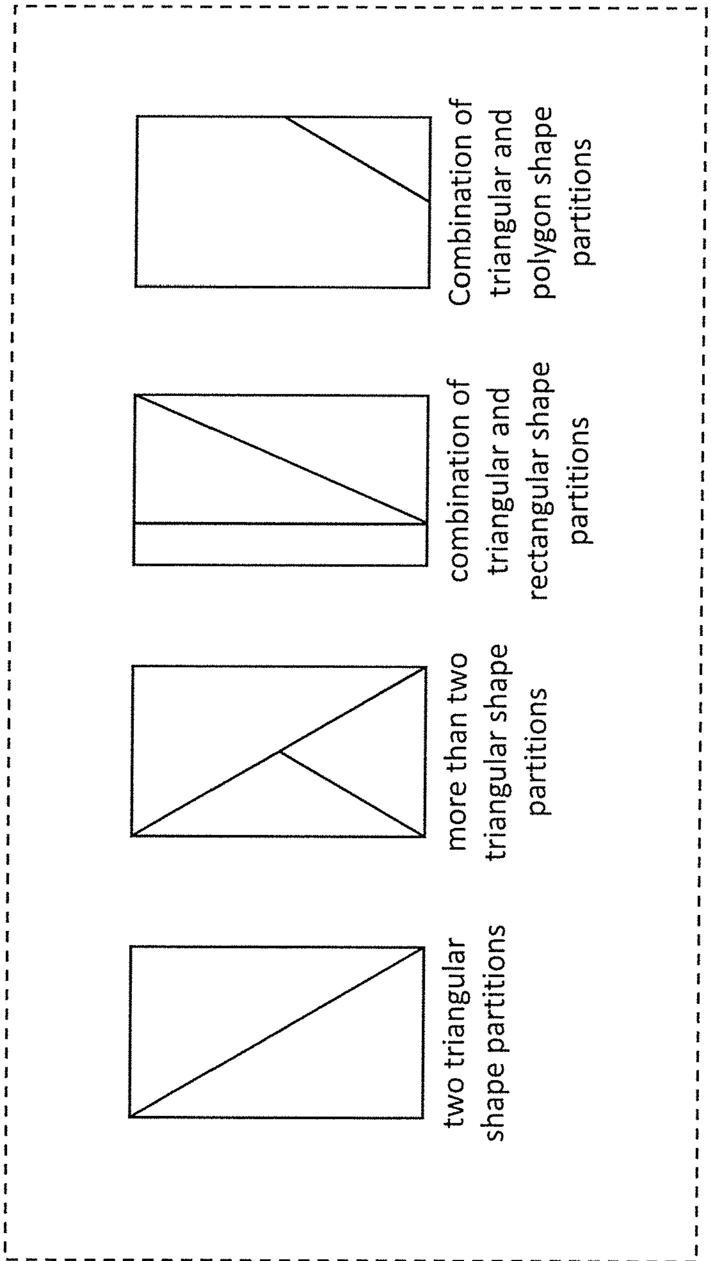
[Fig. 16]

Index	binarization
0	00
1	010
2	0110
3	0111
4	11000
5	11001
6	11010
7	11011
8	11100
9	11101
10	11110
11	11111
12	1010000
⋮	

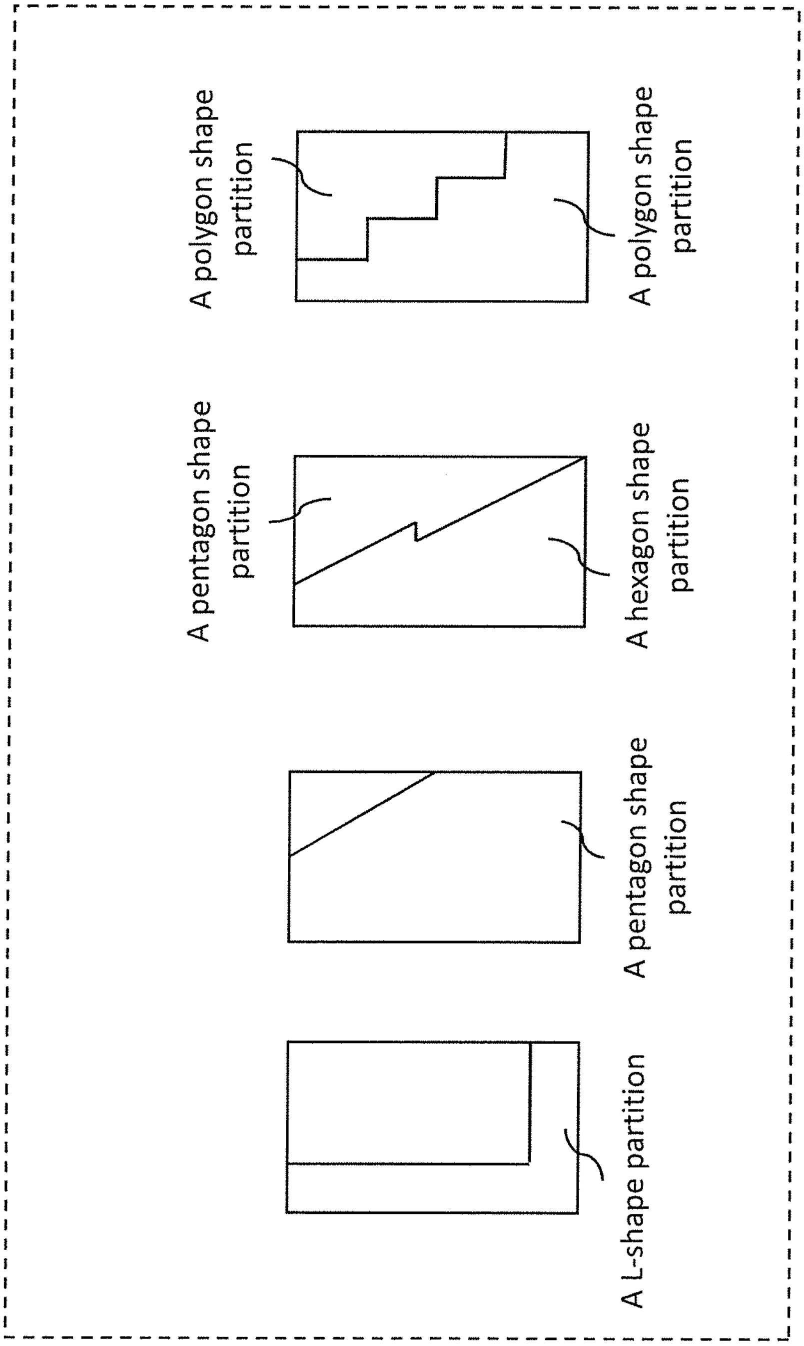
[Fig. 17]


2000
start
S2001
splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, which may or may not have a non-rectangular shape
S2002
predicting a first motion vector for the first partition (e.g., as selected from a first set of motion vector candidates) and a second motion vector for the second partition (e.g., as selected from a second set of motion vector candidates)
S2003
encoding or decoding the first partition using the first motion vector and the second partition using the second motion vector
end

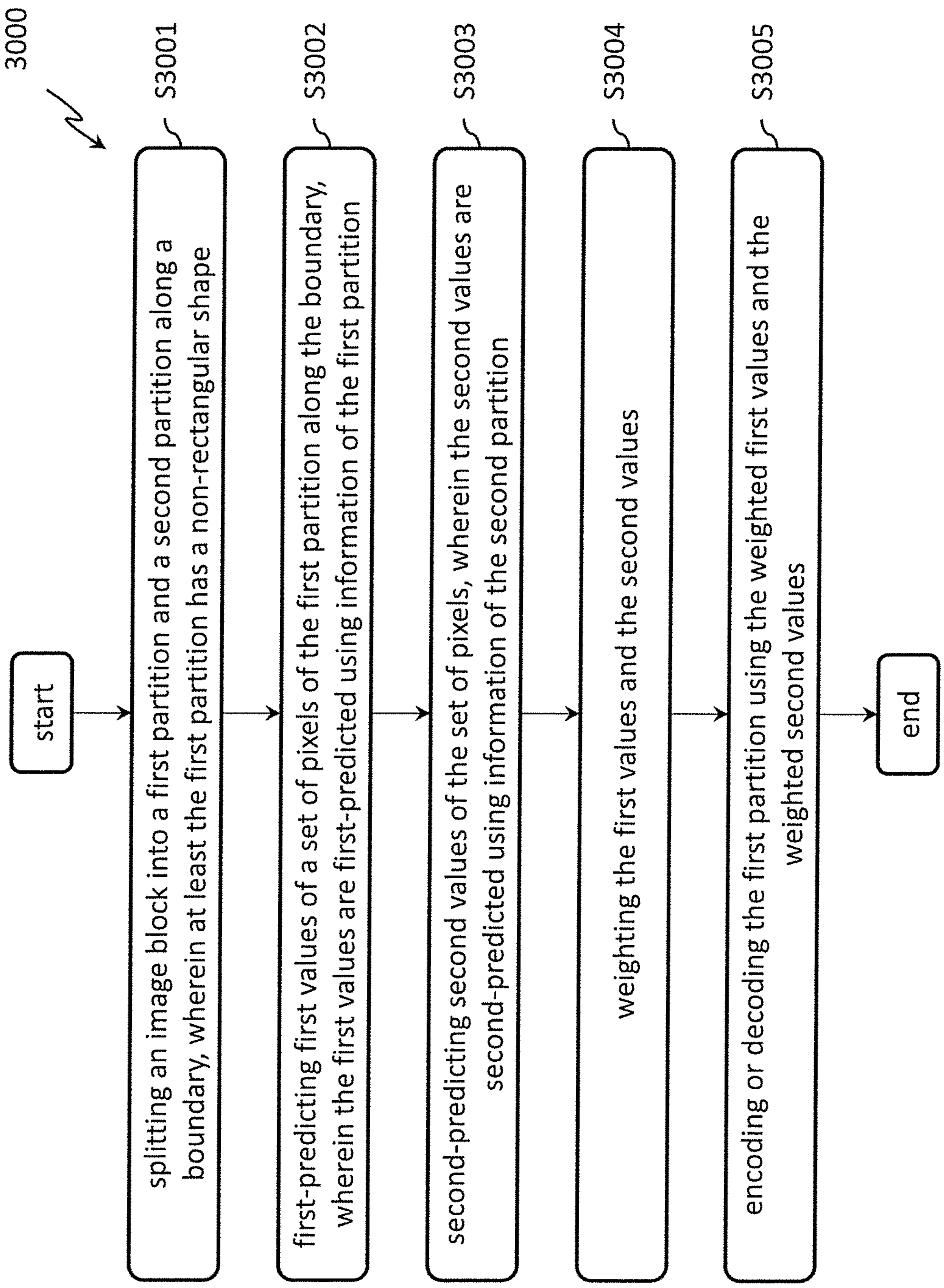
[Fig. 18]


Combination of triangular and polygon shape partitions
combination of triangular and rectangular shape partitions
more than two triangular shape partitions
two triangular shape partitions

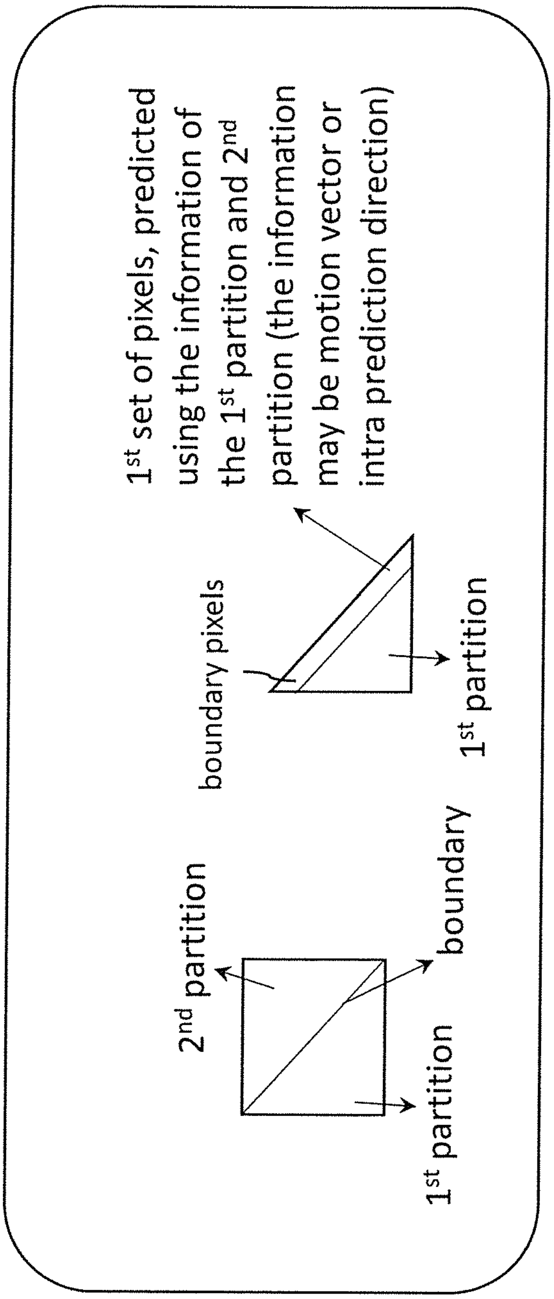
[Fig. 19]


A polygon shape partition
A polygon shape partition
A pentagon shape partition
A hexagon shape partition
A pentagon shape partition
A L-shape partition

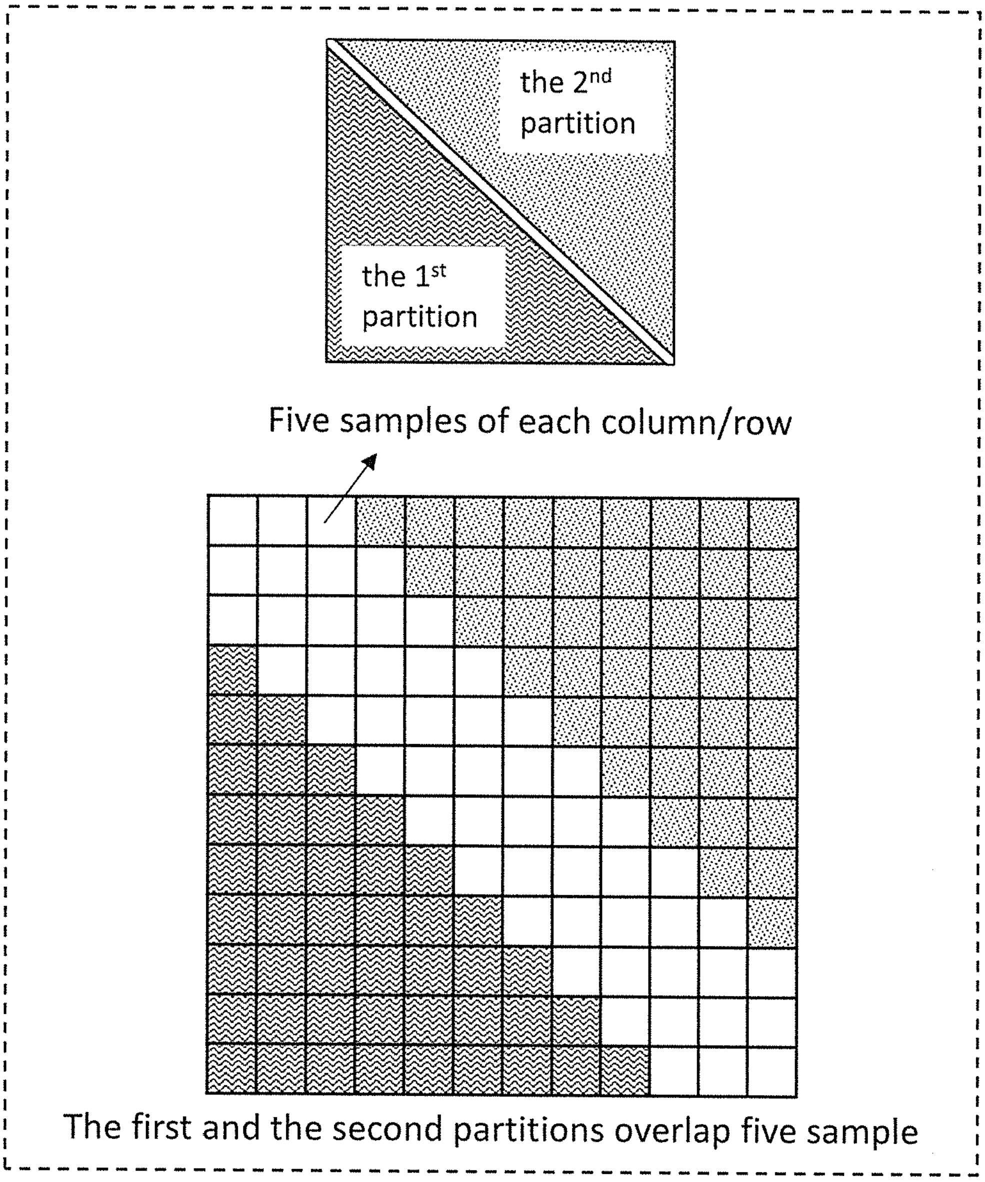
[Fig. 20]


3000
start
S3001
splitting an image block into a first partition and a second partition along a boundary, wherein at least the first partition has a non-rectangular shape
S3002
first-predicting first values of a set of pixels of the first partition along the boundary, wherein the first values are first-predicted using information of the first partition
S3003
second-predicting second values of the set of pixels, wherein the second values are second-predicted using information of the second partition
S3004
weighting the first values and the second values
S3005
encoding or decoding the first partition using the weighted first values and the weighted second values
end

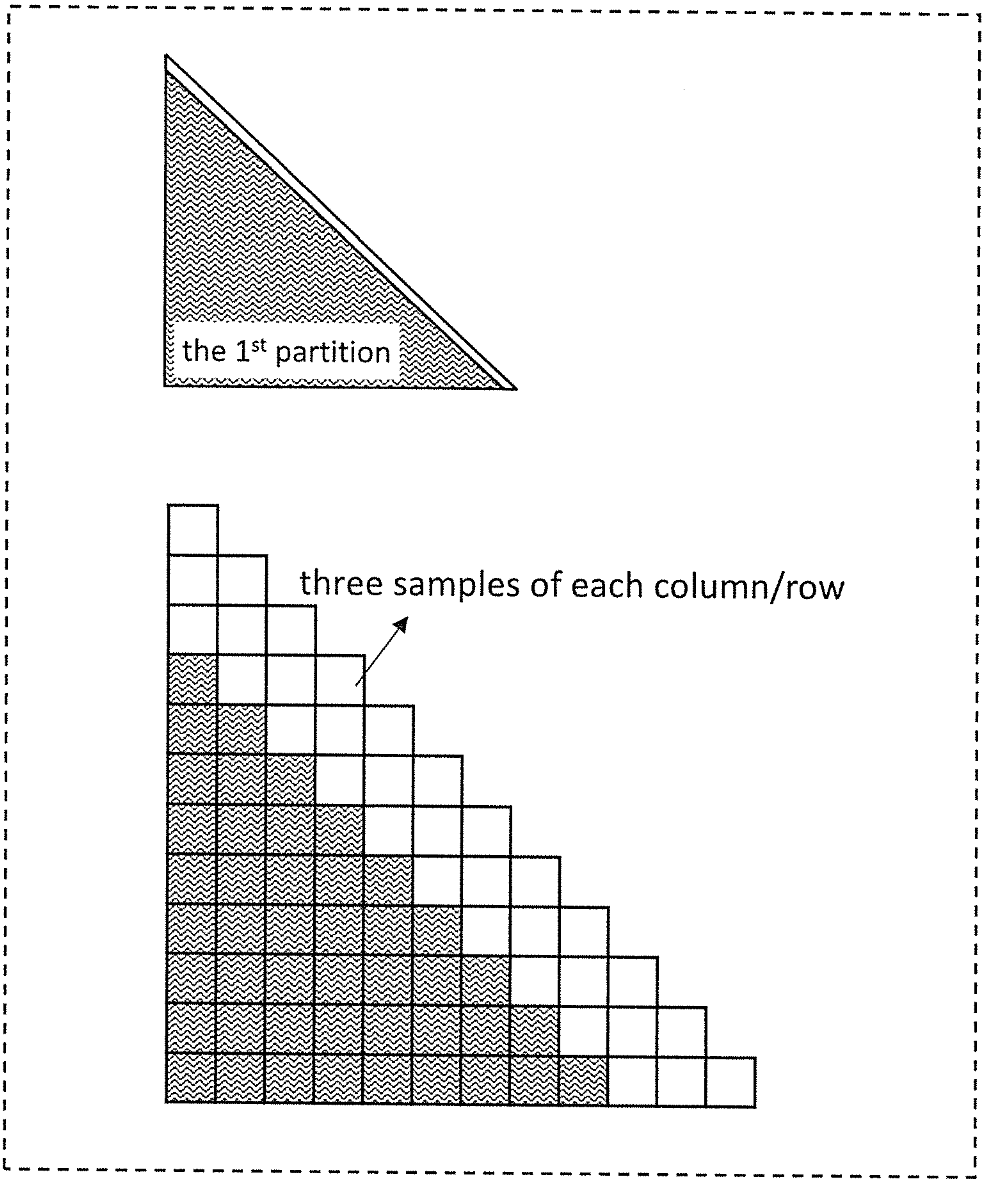
[Fig. 21A]


1st set of pixels, predicted using the information of the 1st partition and 2nd partition (the information may be motion vector or intra prediction direction)
boundary pixels
1st partition
2nd partition
boundary
1st partition

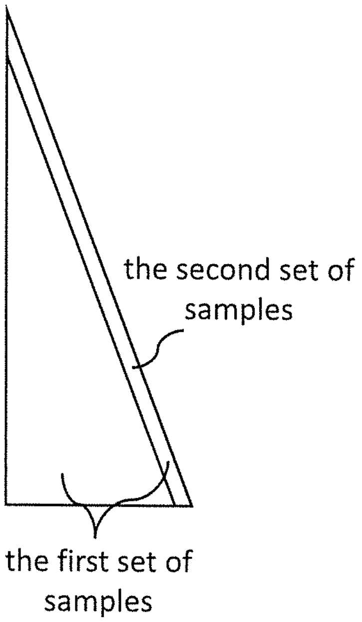
[Fig. 21B]


the 2nd partition
the 1st partition
Five samples of each column/row
The first and the second partitions overlap five sample

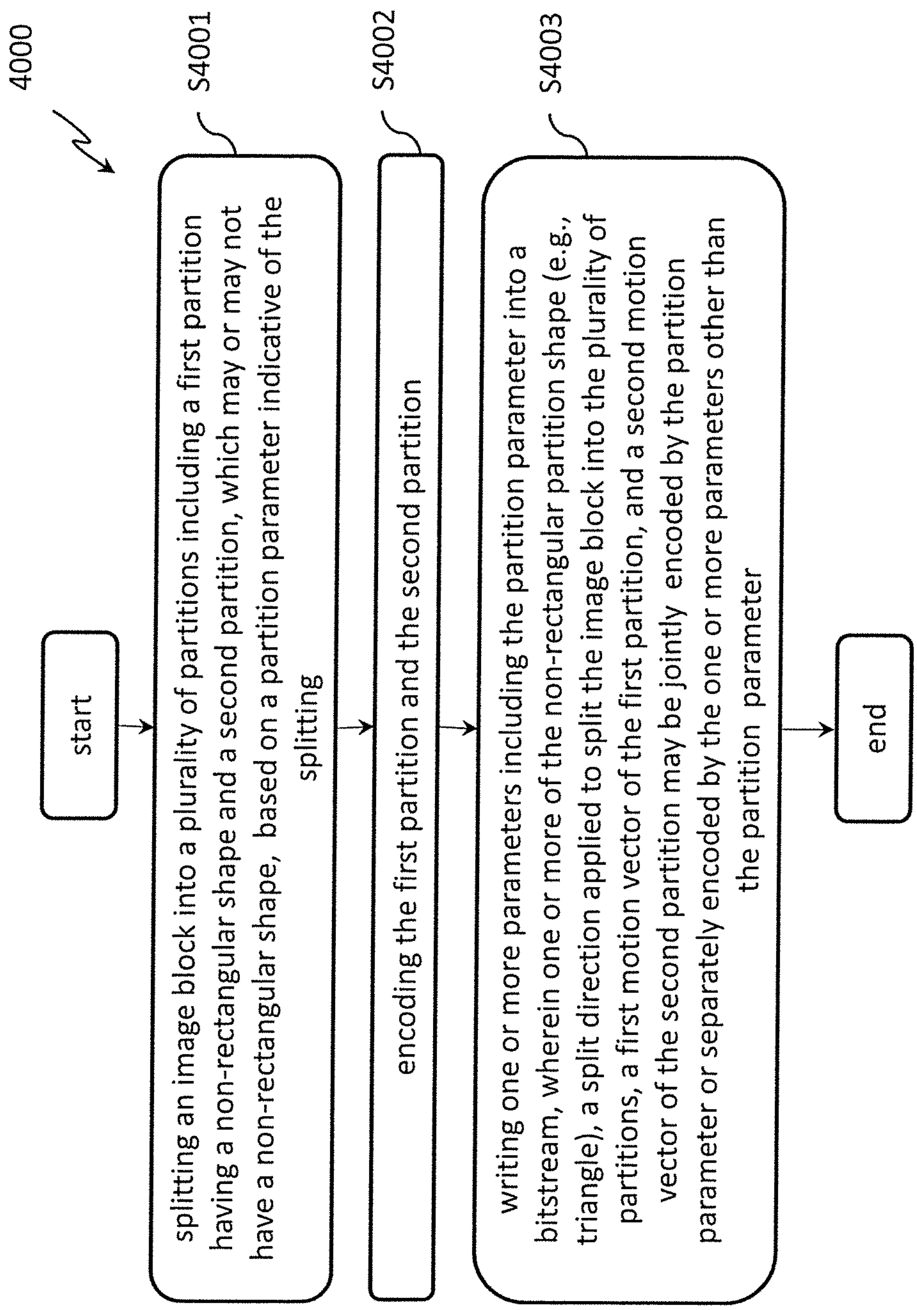
[Fig. 21C]



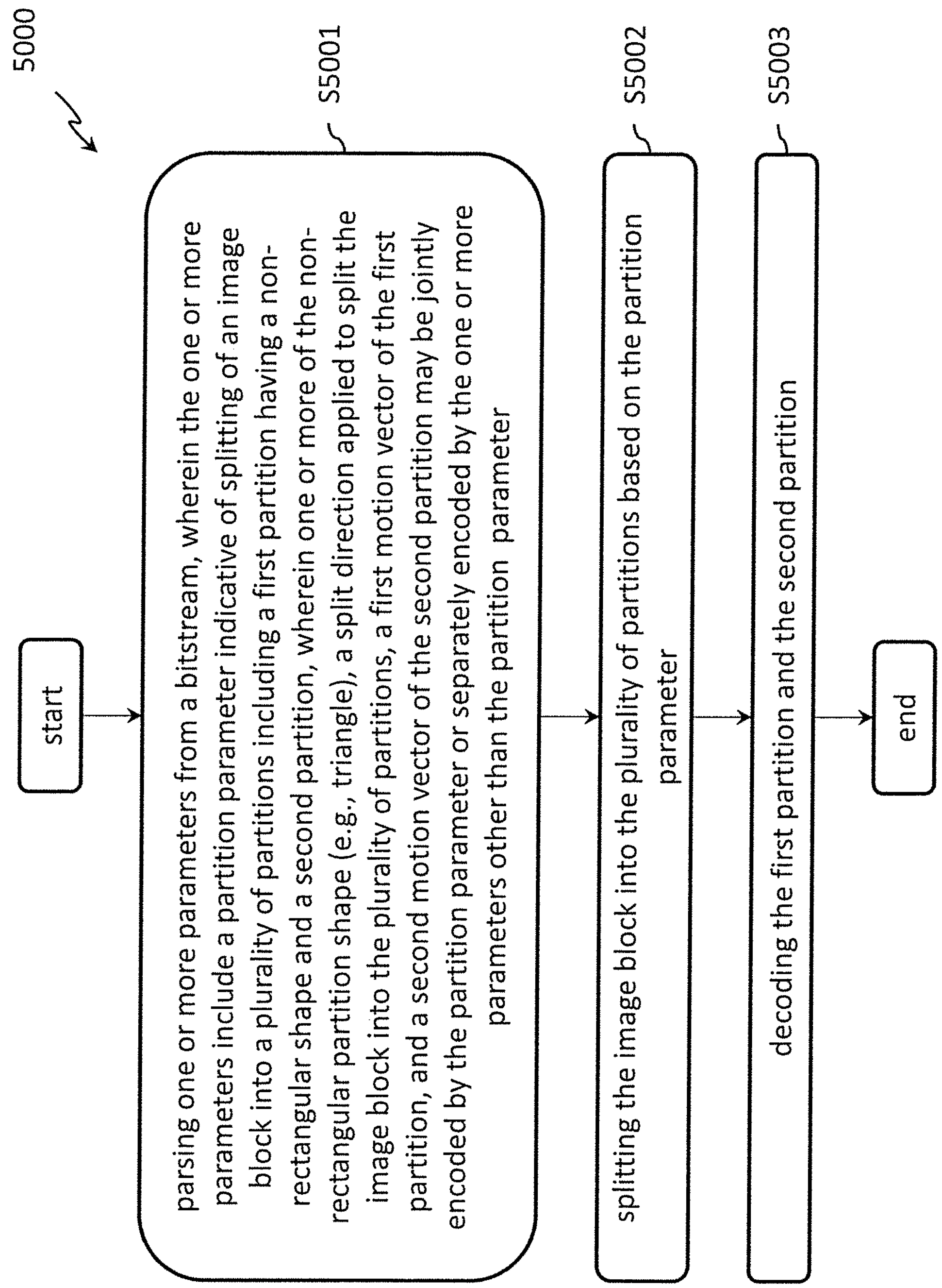
[Fig. 21D]



[Fig. 22]


4000
start
S4001
splitting an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, which may or may not have a non-rectangular shape, based on a partition parameter indicative of the splitting
S4002
encoding the first partition and the second partition
S4003
writing one or more parameters including the partition parameter into a bitstream, wherein one or more of the non-rectangular partition shape (e.g., triangle), a split direction applied to split the image block into the plurality of partitions, a first motion vector of the first partition, and a second motion vector of the second partition may be jointly encoded by the partition parameter or separately encoded by the one or more parameters other than the partition parameter
end

[Fig. 23]


5000
start
S5001
parsing one or more parameters from a bitstream, wherein the one or more parameters include a partition parameter indicative of splitting of an image block into a plurality of partitions including a first partition having a non-rectangular shape and a second partition, wherein one or more of the non-rectangular partition shape (e.g., triangle), a split direction applied to split the image block into the plurality of partitions, a first motion vector of the first partition, and a second motion vector of the second partition may be jointly encoded by the partition parameter or separately encoded by the one or more parameters other than the partition parameter
S5002
splitting the image block into the plurality of partitions based on the partition parameter
S5003
decoding the first partition and the second partition
end

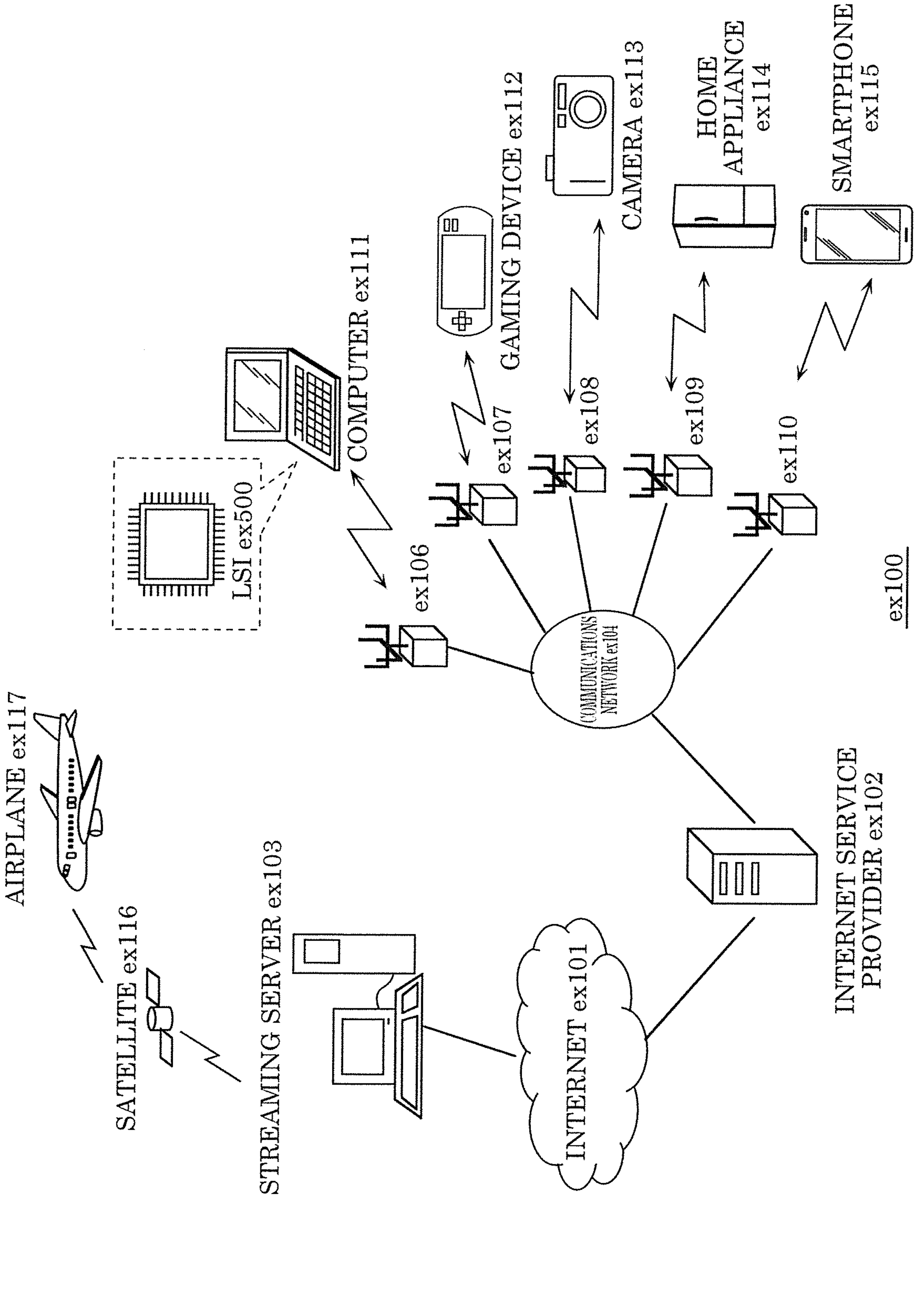
[Fig. 24]

the first index value	the shape of two partitions split from the image block	the direction of splitting the image block into two triangles	the motion vector of the first partition	the motion vector of the second partition
0	not triangular shape	N/A	N/A	N/A
1	triangular shape	From top-left corner to bottom-right corner	the second motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
2	triangular shape	From top-right corner to bottom-left corner	the first motion vector in the first motion vector candidate prediction list	the second motion vector in the second motion vector candidate prediction list
3	triangular shape	From top-right corner to bottom-left corner	the second motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
4	triangular shape	From top-left corner to bottom-right corner	the second motion vector in the first motion vector candidate prediction list	the second motion vector in the second motion vector candidate prediction list
5	triangular shape	From top-right corner to bottom-left corner	the second motion vector in the first motion vector candidate prediction list	the third motion vector in the second motion vector candidate prediction list
6	triangular shape	From top-left corner to bottom-right corner	the third motion vector in the first motion vector candidate prediction list	the first motion vector in the second motion vector candidate prediction list
⋮				

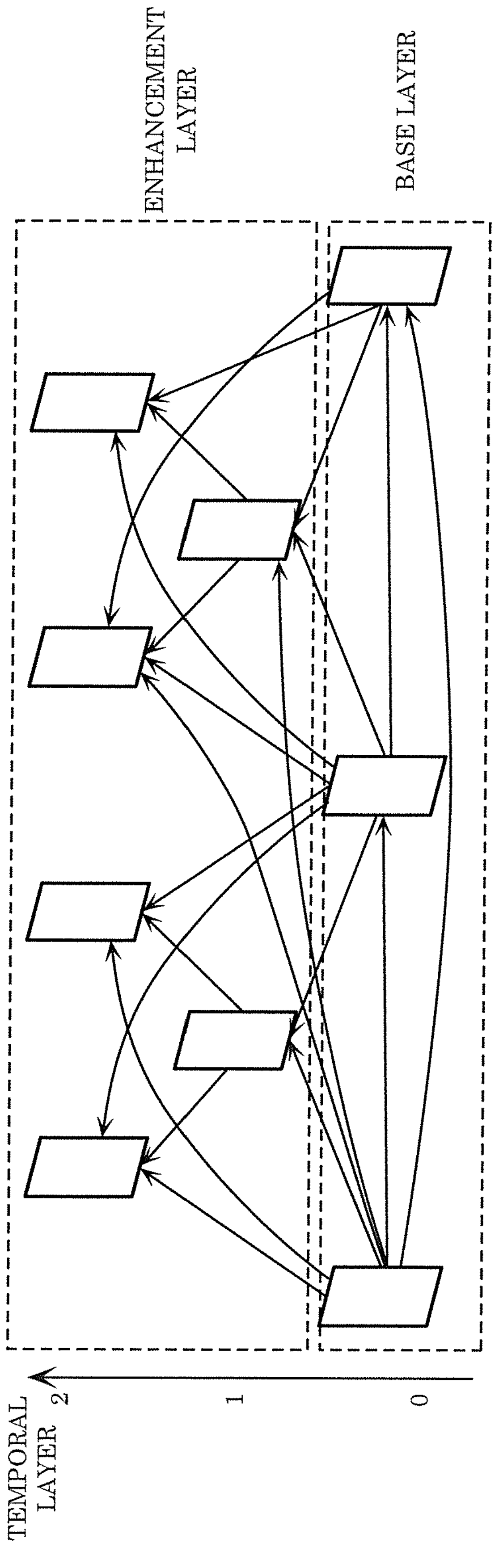
[Fig. 25]

First parameter	Second parameter	Example
block size	a flag to indicate at least one of plural partitions split from an image block is triangular shape	1) When the block size is larger than 64x64, there is no triangular shape partition. 2) When the ratio of width and height is larger than 4 (e.g. 64x4), there is no triangular shape partition.
prediction mode	a flag to indicate at least one of plural partitions split from an image block is triangular shape	1) When an image block is coded in intra mode, there is no triangular shape partition.
a flag to indicate at least one of plural partitions split from an image block is triangular shape	prediction mode	1) When at least one of plural partitions split from an image block is triangular shape, the image block must be inter coded.
the motion vector of neighboring block	the direction of splitting an image block into two triangles	1) The motion vector of neighboring block is diagonal direction, the direction of splitting the image block into two triangles is from top-left corner to bottom-right corner.
the intra prediction direction of neighboring block	the direction of splitting an image block into two triangles	1) The intra prediction direction of neighboring block is inverse-diagonal direction, the direction of splitting the image block into two triangles is from top-right corner to bottom-left corner.

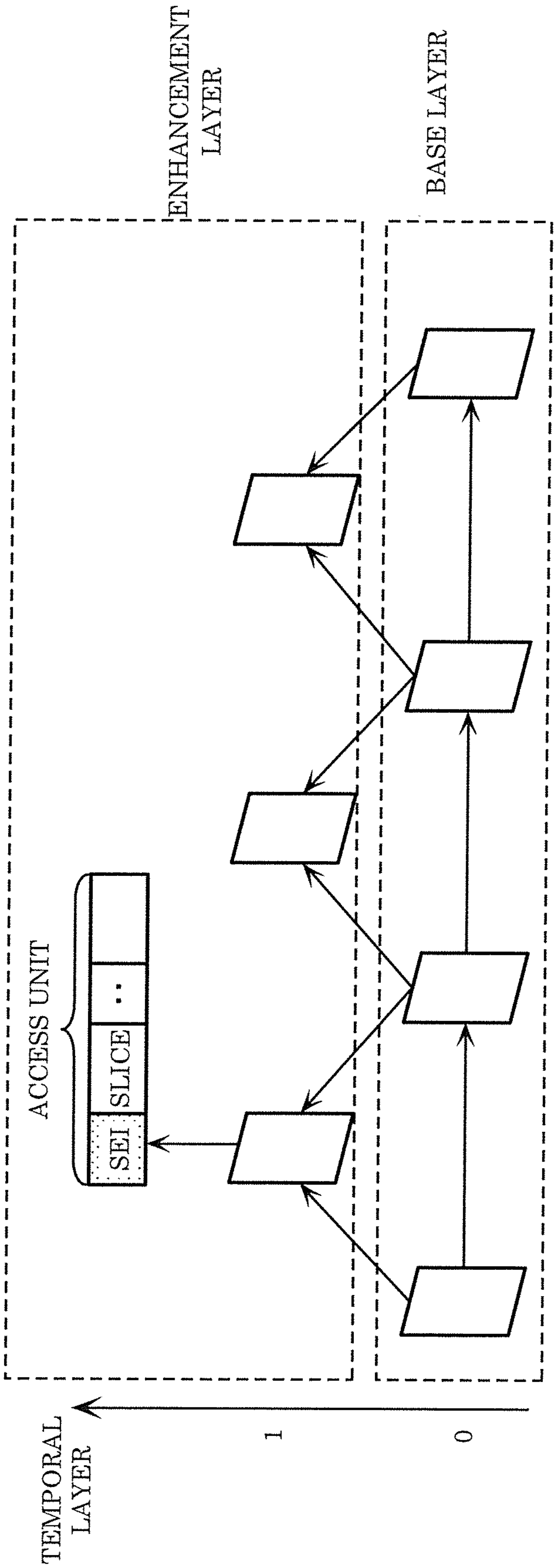
[Fig. 26]


AIRPLANE ex117
SATELLITE ex116
STREAMING SERVER ex103
INTERNET ex101
INTERNET SERVICE PROVIDER ex102
COMMUNICATIONS NETWORK ex104
ex106
ex107
ex108
ex109
ex110
LSI ex500
COMPUTER ex111
GAMING DEVICE ex112
CAMERA ex113
HOME APPLIANCE ex114
SMARTPHONE ex115
ex100

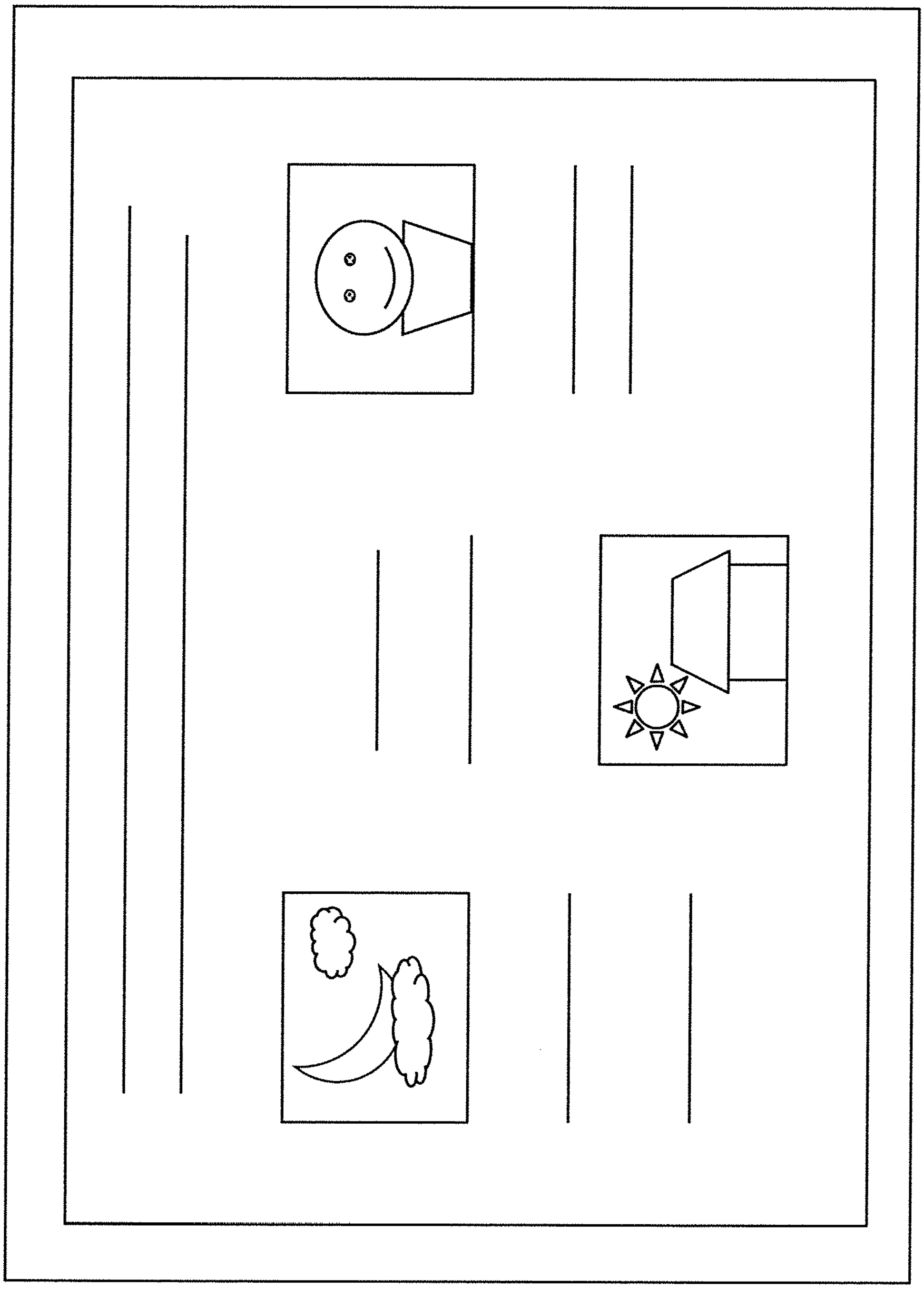
[Fig. 27]



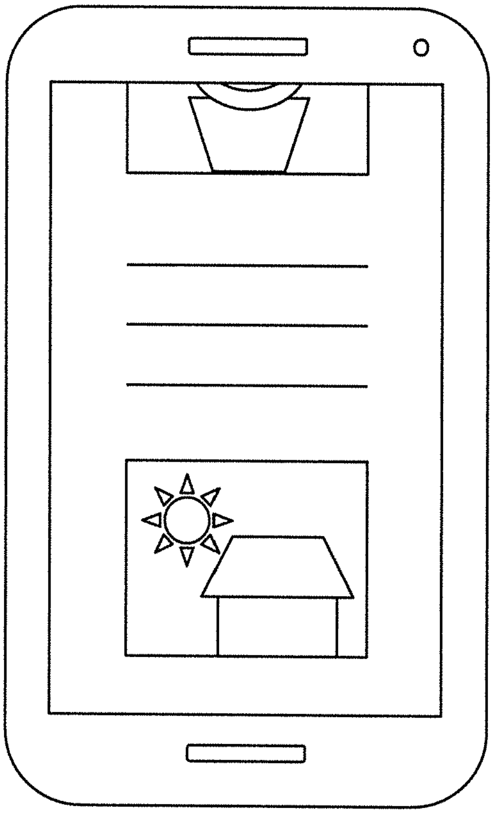
[Fig. 28]


ENHANCEMENT LAYER
BASE LAYER
ACCESS UNIT
SEI
SLICE
TEMPORAL LAYER
1
0

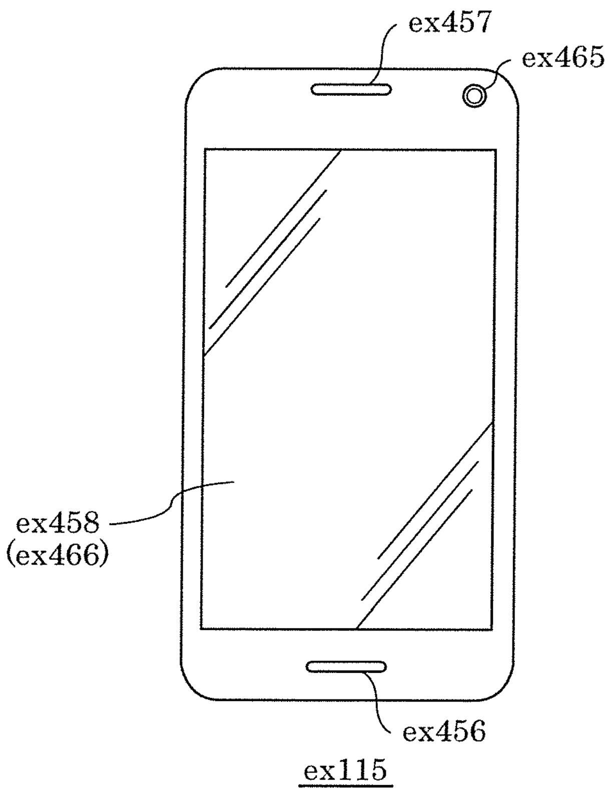
[Fig. 29]



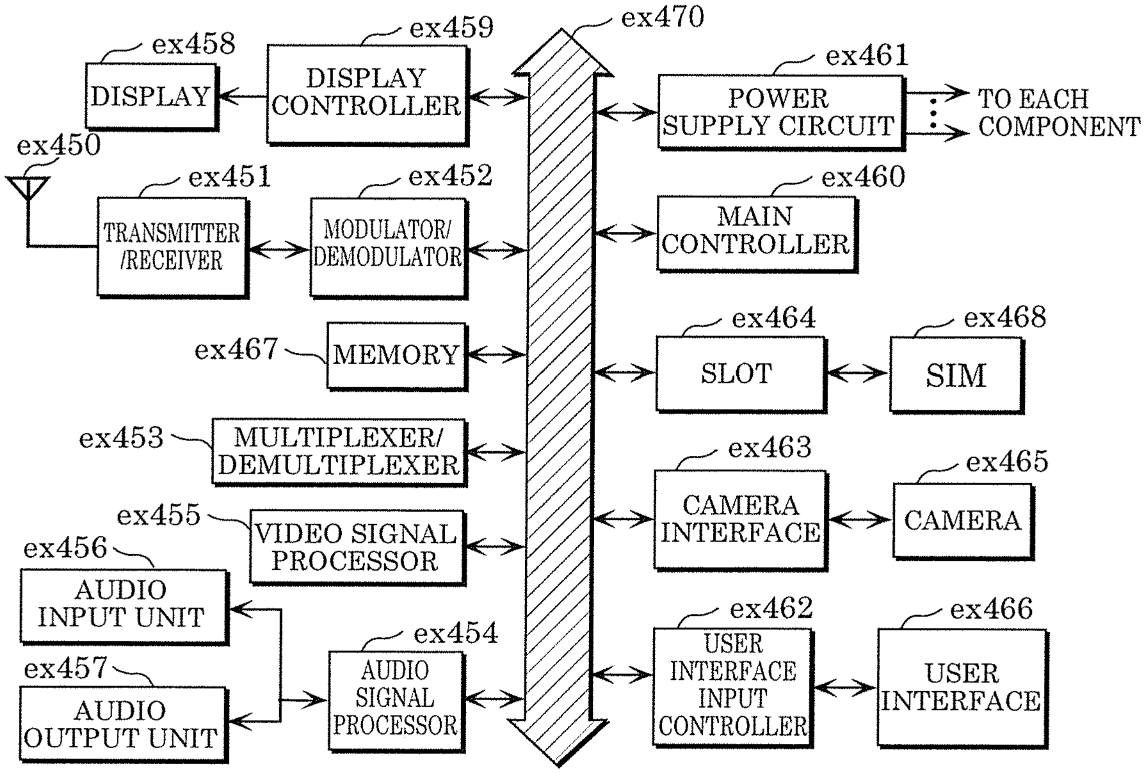
[Fig. 30]



[Fig. 31]



[Fig. 32]


ex458
DISPLAY
ex459
DISPLAY CONTROLLER
ex470
ex461
POWER SUPPLY CIRCUIT
TO EACH COMPONENT
ex450
ex451
TRANSMITTER /RECEIVER
ex452
MODULATOR/ DEMODULATOR
ex460
MAIN CONTROLLER
ex467
MEMORY
ex464
SLOT
ex468
SIM
ex453
MULTIPLEXER/ DEMULTIPLEXER
ex463
CAMERA INTERFACE
ex465
CAMERA
ex455
VIDEO SIGNAL PROCESSOR
ex456
AUDIO INPUT UNIT
ex454
AUDIO SIGNAL PROCESSOR
ex457
AUDIO OUTPUT UNIT
ex462
USER INTERFACE INPUT CONTROLLER
ex466
USER INTERFACE

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2018/030059

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H04N19/537(2014.01)i, H04N19/105(2014.01)i, H04N19/157(2014.01)i, H04N19/96(2014.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. H04N19/537, H04N19/105, H04N19/157, H04N19/96

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2018
Registered utility model specifications of Japan 1996-2018
Published registered utility model applications of Japan 1994-2018

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 2013/0266070 A1 (SONY CORPORATION) 2013.10.10, pars. 0005, 0069-0131, FIGs. 1-15 & JP 2012-23597 A	1-8, 12-26, 30-35 9-10, 27-28
Y	WO 2015/006884 A1 (QUALCOMM INCORPORATED) 2015.01.22, pars. 0001-0030, 0180-0200 FIGs. 1-11 (No Family)	9-10, 27-28
A	US 2016/0234503 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 2016.08.11, Whole Document & JP 2016-533667 A	1-35
A	US 2008/0101707 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 2008.05.01, Whole Document & JP	1-35

☑ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16.10.2018	23.10.2018

Name and mailing address of the ISA/JP
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer 5C 6297
USHIMARU, Taiki
Telephone No. +81-3-3581-1101 Ext. 3541

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	2010-508706 A	
A	Xiaozhen Zheng et al., CE2: Test results of non-rectangular motion partitioning (NRMP) with overlapped block motion compensation (OBMC), Joint Collaborative Team on Video Coding (JCT-VC) 5th Meeting: Geneva, JCTVC-E373.doc, JCTVC-E373-v1.zip, 2011.03.16	1-35
E,A	Tadamasa Toma et al., Description of SDR video coding technology proposal by Panasonic, Joint Video Experts Team (JVET) 10th Meeting: San Diego, JVET-J0020-v1.docx, JVET-J0020-v2.zip, 2018.04.20	1-35