



(51) International Patent Classification:
H04N 7/32 (2006.01)

(21) International Application Number:
PCT/JP2012/001318

(22) International Filing Date:
27 February 2012 (27.02.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2011-051269 9 March 2011 (09.03.2011) JP

(71) Applicant (for all designated States except US): **CANON KABUSHIKI KAISHA** [JP/JP]; 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **KOBAYASHI, Masaaki** [JP/JP]; C/O CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (JP).

(74) Agents: **ABE, Takuma** et al.; C/O CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, 1468501 (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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.

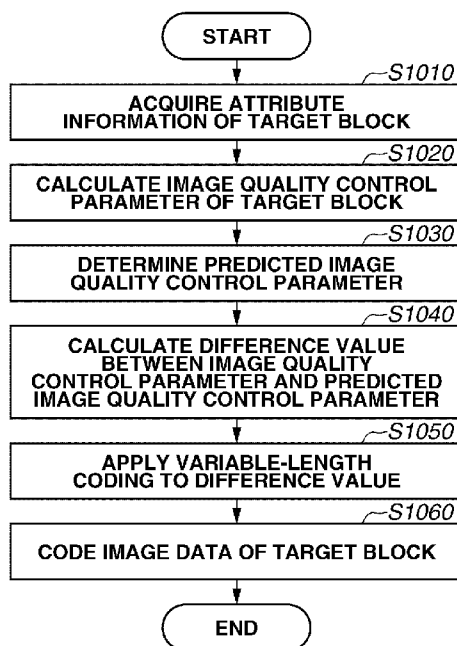
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, 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, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: IMAGE CODING APPARATUS, METHOD FOR CODING IMAGE, PROGRAM THEREFOR, IMAGE DECODING APPARATUS, METHOD FOR DECODING IMAGE, AND PROGRAM THEREFOR

[Fig. 1]



(57) Abstract: An image coding method for dividing an input image into a plurality of divisional blocks having different sizes, and coding the image for each of the divisional blocks includes acquiring attribute information of a target block, setting a control parameter for controlling an image quality of the target block, determining a prediction control parameter based on the attribute information, calculating a difference value between the control parameter and the prediction control parameter, and coding the calculated difference value.

Description

Title of Invention: IMAGE CODING APPARATUS, METHOD FOR CODING IMAGE, PROGRAM THEREFOR, IMAGE DECODING APPARATUS, METHOD FOR DECODING IMAGE, AND PROGRAM THEREFOR

Technical Field

[0001] The present invention relates to an image coding apparatus, a method for coding an image, a program therefor, an image decoding apparatus, a method for decoding an image, and a program therefor. More particularly, the present invention relates to a method for coding an image quality control parameter and a method for decoding an image quality control parameter.

Background Art

[0002] As a method for compressing and recording a moving image, MPEG-2 Video (hereafter referred to simply as MPEG-2) (ISO/IEC 13818-2:2000 Information technique - Generic coding of moving pictures and associated audio information: Video) and H.264 (ISO/IEC 14496-10:2004 Information technique - Coding of audio-visual targets - Part 10: Advanced Video Coding) are known. In recent years, the Joint Collaborative Team on Video Coding (JCT-VC) has been established as a collegial organization of the ITU-T and ISO/IEC. This organization advances standardization activities for the High Efficiency Video Coding (HEVC) which is a new moving image standard. For example, in a JCT-VC contribution JCTVC-A205.doc (http://wftp3.itu.int/av-arch/jctvc-site/2010_04_A_Dresden/), HEVC proposes an H.264-based improved technique.

[0003] With a coding method based on the orthogonal transform and quantization represented by MPEG-2, H.264, and HEVC, the coding side applies orthogonal transform and quantization to a predetermined block image to generate quantization coefficient data. In this case, the coding method controls the image quality by quantizing the image data by using an image quality control parameter which is called quantization parameter. Specifically, quantization by using a small quantization parameter value improves the image quality and increases the amount of codes, and quantization by using a large quantization parameter value degrades the image quality and reduces the amount of codes. In the coding processing, the coding method codes the image data while selecting an optimum quantization parameter value according to a target amount of codes in this way. This method is referred to as rate control with which TM5 (MPEG-2 Test Model 5 (TM5), Doc. ISO/IEC JTC1/SC29/WG11/N0400, Test Model

Editing Committee, April 1993) and other various methods have been proposed. Japanese Patent Application Laid-Open No. 2001-45494 discusses a technique for determining the visual importance of an image area and controlling an image quality control parameter according to the importance.

- [0004] Variable-length coding is applied to quantized quantization coefficient data to generate variable-length coding coefficient data. The quantization parameter is also coded to generate a quantization parameter code. For example, a quantization parameter generation method uses as a predictive quantization parameter a quantization parameter used to quantize a block quantized before a block subjected to quantization, and calculates a difference value between the predictive quantization parameter and a quantization parameter used to quantize the block subjected to quantization. This difference value, called Quantization Parameter Delta (QP_DELTA), is embedded in a bit stream as a quantization parameter code. The variable-length coding coefficient data and the quantization parameter code generated in this way are transmitted as a bit stream to a decoding unit via an optical disk medium or a network. The decoding side decodes the variable-length coding coefficient data and the quantization parameter code to generate quantization coefficient data and a quantization parameter, and applies inverse orthogonal transform and inverse quantization to the quantization coefficient data by using the quantization parameter to generate a decoded image.
- [0005] With MPEG-2 and H.264, processing is performed in unit of a lattice block formed of 16x16 pixels formed by dividing an image called a macroblock. When the size of a block subjected to orthogonal transform is represented in pixel units, the block size is 8x8 pixels with MPEG-2, or 8x8 pixels or 4x4 pixels with H.264. Specifically, one macroblock includes a plurality of orthogonal transform blocks. Since MPEG-2 or H.264 enables controlling the quantization parameter in macroblock units (rate control), orthogonal transform blocks included in an identical macroblock are quantized based on a similar quantization parameter.
- [0006] With HEVC, on the other hand, lattice blocks formed by dividing an image in lattice form are called Largest Coding Units (LCUs) each being formed of 64x64 pixels. Each LCU is divided into a plurality of smaller-sized blocks called Coding Units (CUs) by using the region quadtree structure. Each CU includes orthogonal transform blocks called Transform Units (TUs). Each TU is further divided into a plurality of smaller-sized blocks by using the region quadtree structure. Each unit of a quantization parameter has a division flag. A block having a True division flag includes four divisional blocks each being horizontally and vertically half in size of the block. A block having a False division flag does not include divisional blocks but has actual data of the block. There are various methods for determining whether a block is to be divided.

As a method for determining whether a block is to be divided, Japanese Patent Application Laid-Open No. 2005-191706 discusses a technique for calculating a cost of blocks by using the Lagrange multiplier and selecting a block division method with a lowest cost.

[0007] With a quantization parameter coding method for coding as a quantization parameter code a difference value between a quantization parameter and a predictive quantization parameter of a block subjected to coding, the amount of quantization parameter codes increases with increasing absolute value of the difference value. When embedding a quantization parameter in block units in a conventional case, since there is only one method for calculating a predictive quantization parameter, the absolute value of the difference value increases, thus resulting in an unnecessarily increased amount of quantization parameter codes depending on the image coding method and image characteristics.

Summary of Invention

[0008] The present invention is directed to preventing an unnecessary increase in the amount of quantization parameter codes and to reducing the amount of quantization parameter codes.

[0009] According to an aspect of the present invention, an image coding method for dividing an input image into a plurality of divisional blocks having different sizes, and coding the image for each of the divisional blocks includes: acquiring attribute information of a target block; setting a control parameter for controlling an image quality of the target block; determining a prediction control parameter based on the attribute information; calculating a difference value between the control parameter and the prediction control parameter; and coding the calculated difference value.

[0010] According to an exemplary embodiment of the present invention, the amount of quantization parameter codes can be reduced.

[0011] Further features and aspects of the present invention will become apparent from the following detailed description of exemplary embodiments with reference to the attached drawings.

Brief Description of Drawings

[0012] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate exemplary embodiments, features, and aspects of the invention and, together with the description, serve to explain the principles of the invention.

[0013] [fig.1] Fig. 1 is a flowchart illustrating a divisional block coding method according to a first exemplary embodiment of the present invention.

[fig.2] Fig. 2 is a flowchart illustrating a divisional block coding method according to a third exemplary embodiment of the present invention.

[fig.3]Fig. 3 is a flowchart illustrating a divisional block coding method according to a fifth exemplary embodiment of the present invention.

[fig.4]Fig. 4 is a flowchart illustrating a divisional block coding method according to a seventh exemplary embodiment of the present invention.

[fig.5]Fig. 5 is a flowchart illustrating a divisional block coding method according to a ninth exemplary embodiment of the present invention.

[fig.6]Fig. 6 is a flowchart illustrating a divisional block decoding method according to a second exemplary embodiment of the present invention.

[fig.7]Fig. 7 is a flowchart illustrating a divisional block decoding method according to a fourth exemplary embodiment of the present invention.

[fig.8]Fig. 8 is a flowchart illustrating a divisional block decoding method according to a sixth exemplary embodiment of the present invention.

[fig.9]Fig. 9 is a flowchart illustrating a divisional block decoding method according to an eighth exemplary embodiment of the present invention.

[fig.10]Fig. 10 is a flowchart illustrating a divisional block decoding method according to a tenth exemplary embodiment of the present invention.

[fig.11A]Fig. 11A illustrates a hardware configuration according to exemplary embodiments of the present invention.

[fig.11B]Fig. 11B illustrates a hardware configuration according to exemplary embodiments of the present invention.

[fig.11C]Fig. 11C illustrates a hardware configuration according to exemplary embodiments of the present invention.

[fig.12A]Fig. 12A illustrates a relation between positions of neighboring blocks.

[fig.12B]Fig. 12B illustrates a relation between positions of neighboring blocks.

[fig.13A]Fig. 13A illustrates example block division.

[fig.13B]Fig. 13B illustrates example block division.

[fig.14]Fig. 14 is a table illustrating relations between block positions, and identification value conversion.

[fig.15]Fig. 15 is a flowchart illustrating a method for coding a frame.

[fig.16]Fig. 16 is a flowchart illustrating a method for decoding a frame.

[fig.17]Fig. 17 illustrates an example configuration of a bit stream of a frame.

[fig.18A]Fig. 18A illustrates an example configuration of bit streams of lattice blocks.

[fig.18B]Fig. 18B illustrates an example configuration of bit streams of lattice blocks.

[fig.19A]Fig. 19A illustrates an example division state of lattice blocks and division flags.

[fig.19B]Fig. 19B illustrates an example division state of lattice blocks and division flags.

[fig.19C]Fig. 19C illustrates an example division state of lattice blocks and division

flags.

[fig.19D]Fig. 19D illustrates an example division state of lattice blocks and division flags.

[fig.20]Fig. 20 is a flowchart illustrating a predicted image quality control parameter acquisition method for each block size of a divisional block.

[fig.21]Fig. 21 is a flowchart illustrating processing for determining a predicted image quality control parameter based on an intra-prediction method.

[fig.22]Fig. 22 illustrates referential relations between image quality control parameters of respective blocks.

[fig.23A]Fig. 23A illustrates an example operation of the divisional block coding method according to the fifth exemplary embodiment of operation.

[fig.23B]Fig. 23B illustrates an example operation of the divisional block coding method according to the fifth exemplary embodiment of operation.

[fig.23C]Fig. 23C illustrates an example operation of the divisional block coding method according to the fifth exemplary embodiment of operation.

[fig.23D]Fig. 23D illustrates an example operation of the divisional block coding method according to the fifth exemplary embodiment of operation.

[fig.24A]Fig. 24A illustrate an example predicted image quality control parameter acquisition method.

[fig.24B]Fig. 24B illustrate the example predicted image quality control parameter acquisition method.

[fig.25]Fig. 25 is illustrates an index indicating prediction directions in intra-prediction.

Description of Embodiments

[0014] Various exemplary embodiments, features, and aspects of the invention will be described in detail below with reference to the drawings.

[0015] A method for coding a block subjected to coding according to a first exemplary embodiment of the present invention will be described below with reference to Figs. 1, 11A, 11B, 11C, 15, and 20.

[0016] Fig. 11A illustrates a hardware configuration for executing a coding program including the coding method according to the first exemplary embodiment. This coding program is recorded on a hard disk drive (hereinafter referred to as HDD) 1104. When the coding program is activated, it is loaded into a read-only memory (RAM) 1102, and a central processing unit (CPU) 1101 executes each step of the coding program described below to perform coding processing. In the present exemplary embodiment, input image data is stored in the HDD 1104 and, a coded bit stream is also recorded on the HDD 1104. Coding processing with the above-mentioned configuration will be

described below. In the present exemplary embodiment, a frame is divided into a plurality of lattice blocks each being formed of 64x64 pixels (hereinafter, a block formed of nxn pixels is referred to as an nxn block). Each lattice block is further divided, by using the region quadtree structure, into a plurality of divisional blocks each being formed of at least 8x8 pixels. Division blocks are subjected to coding processing. Fig. 13A illustrates an example state where a frame is divided into a plurality of lattice blocks. Fig. 13B illustrates an example state where a lattice block is further divided into a plurality of divisional blocks. Even when a lattice block is left undivided, processing is performed on the premise that the lattice block includes one 64x64 divisional block.

- [0017] Fig. 15 is a flowchart illustrating processing for coding the entire frame. In step S15010, the CPU 1101 codes the frame header information. The frame header information includes a frame synchronization code, and frame type information indicating whether the frame is an intra-frame or an inter-frame.
- [0018] In step S15020, the CPU 1101 sets a lattice block subjected to processing (hereinafter referred to as target lattice block). A target lattice block subjected to the first processing is a lattice block at the top left corner of a frame. Subsequently, each time step S15020 is executed, a lattice block is set in order of raster scanning, for example, in order of lattice block numbers (B1, B2, ...) illustrated in Fig. 13A.
- [0019] In step S15030, the CPU 1101 determines a division state of the target lattice block and a prediction mode (described below) of each divisional block therein. The target lattice block is divided into a plurality of divisional blocks by using the region quadtree structure, as illustrated in Fig. 13B. In the present exemplary embodiment, the prediction mode indicates an intra-prediction mode and an inter-prediction mode. The intra-prediction mode is a mode in which an intra-predicted image block is generated referring to neighboring divisional blocks of a divisional block subjected to processing (hereinafter referred to as target divisional block) as is the case with H.264, and the inter-prediction mode is a mode in which a motion compensation predicted image block is generated by determining a motion vector.
- [0020] In step S15031, the CPU 1101 codes prediction mode information and divisional block information of divisional blocks in the target lattice block. The prediction mode information includes a prediction mode flag and additional information about the prediction mode. For example, in the intra-prediction mode, the additional information is intra-prediction method information. In the inter-prediction mode, the additional information is motion vector information or motion vector derivation block identification information for identifying a neighboring block having a similar motion vector as the target divisional block. For example, the motion vector derivation block identification information is composed of a 1-bit motion vector identification flag and a motion

vector prediction direction flag. When the motion vector identification flag is 1, the motion vector prediction direction flag is embedded but a motion vector is not. Otherwise, when the motion vector identification flag is 0, a motion vector is embedded but the motion vector prediction direction flag is not. Further, when the motion vector prediction direction flag is 1, a motion vector of the left block is considered as a motion vector of the target block. Otherwise, when the motion vector prediction direction flag is 0, a motion vector of the upper block is considered as a motion vector of the target block. When embedding the motion vector derivation block identification information, a motion vector can be derived from the motion vector derivation block identification information in decoding processing without embedding the motion vector information in a bit stream. Although, in the present exemplary embodiment, the motion vector prediction direction flag is handled as a 1-bit flag, the flag configuration is not limited thereto. The motion vector prediction direction flag may be a multivalued flag to enable selection from a plurality of block candidates including the upper left and upper right blocks, etc., in addition to the upper and left blocks. Similar to the prediction mode with the 16x16 intra-prediction block in H.264, the intra-prediction method information is used to identify, for example, horizontal prediction, vertical prediction, DC prediction, and planar prediction.

[0021] A prediction mode information code, i.e., variable-length coded prediction mode information, is embedded in a bit stream. The divisional block information is information about a division state indicating whether each node (block) of the region quadtree structure is to be further divided into smaller divisional blocks. The divisional block information for a certain node (block) is considered as a division flag. For example, when a lattice block is divided into divisional blocks as illustrated in Fig. 13B, division states for different block sizes are as illustrated in Figs. 19A, 19B, 19C, and 19D. Figs. 19A, 19B, 19C, and 19D illustrate division states of 64x64, 32x32, 16x16, and 8x8 blocks, respectively. The division state (1) indicates that the block is further divided into smaller blocks, and the division state (0) indicates that the block is left undivided. Reference numerals N1, N2, ..., N21 indicate node (block) numbers for the region quadtree structure. These node numbers set the order of scanning the region quadtree structure by depth priority search in each layer, and the scanning order of the upper left, upper right, lower left, and lower right blocks within an identical block (hereinafter referred to as the order of the region quadtree structure). "-" indicates that no node (block) exists. By transmitting as divisional block information the division flags of node numbers N1, N2, ..., N21 to the decoding unit, the region quadtree structure can be reconstructed also on the side of the decoding unit. The prediction mode information code and division flag are embedded in a bit stream for each divisional block, for example, in a structure as illustrated in Fig. 18A. Referring to Fig.

18A, node numbers N1, N2, ..., N21 correspond to node (block) numbers in Figs. 19A, 19B, 19C, and 19D, and reference numerals b1, b2, ..., b16 correspond to divisional block numbers in Fig. 13B. With node numbers N1, N2, ..., N21, the prediction mode information is added only to blocks no longer dividable, i.e., divisional blocks corresponding to reference numerals b1, b2, ..., b16.

[0022] In step S15040, the CPU 1101 sets a target divisional block. At the first time, the CPU 1101 sets the upper left block in the lattice block. Subsequently, each time step S15040 is executed, the CPU 1101 sets a divisional block in order of the region quadtree structure, for example, in order of the divisional block numbers b1, b2, ..., b16 illustrated in Fig. 13B.

[0023] In step S15050, the CPU 1101 codes the target divisional block set in step S15040. Detailed processing in this step will be described below. In step S15060, the CPU 1101 determines whether processing is completed for all divisional blocks in the target lattice block. When processing is completed for all divisional blocks in the target lattice block (YES in step S15060), the processing proceeds to S15070. Otherwise, when processing is not completed for all divisional blocks in the target lattice block (NO in step S15060), the processing returns to step S15040. When processing is completed for all divisional blocks for one lattice block, a bit stream of the lattice block, having a structure as illustrated in Fig. 18A, is generated.

[0024] In step S15070, the CPU 1101 determines whether processing is completed for all lattice blocks in the frame. When processing is completed for all lattice blocks in the frame (YES in step S15070), the processing exits the flowchart in Fig. 15. Otherwise, when processing is not completed for all lattice blocks in the frame (NO in step S15070), the processing returns to step S15020. When processing is completed for all lattice blocks in the frame, a bit stream of the frame, having a structure as illustrated in Fig. 17A, is generated.

[0025] Fig. 1 is a detailed flowchart illustrating the divisional block coding method executed in step S15050 in Fig. 15.

[0026] In step S1010, the CPU 1101 acquires attribute information of the target divisional block. Although, in the present exemplary embodiment, the block size is used as attribute information, the attribute information is not limited thereto. For example, the attribute information may be block mode information for identifying intra- and inter-prediction, a motion vector, or an intra-prediction method in intra-prediction. Using these pieces of attribute information alone or in combination, an image quality control parameter of a last coded divisional block having a similar attribute as the target divisional block may be used as a predicted image quality control parameter. The motion vector does not need to be identical, and may be regarded as identical if an absolute value of a vector difference is equal to or less than a fixed value.

- [0027] In step S1020, the CPU 1101 calculates an image quality control parameter of the target divisional block. The image quality control parameter is calculated so that, for example, the amount of frame codes becomes a preset target value. In this case, the image quality control parameter can be adjusted so that each region has a different image quality control parameter aiming at improving the image quality of a region having specific image components, such as a human face, an object edge, etc.
- [0028] In step S1030, the CPU 1101 acquires an image quality control parameter of a last coded divisional block having a similar attribute as the target divisional block, as a prediction control parameter. In the present exemplary embodiment, since the block size is used as attribute information, the CPU 1101 records, updates, and refers to the image quality control parameter for each block size. Detailed processing in this step will be described below.
- [0029] In step S1040, the CPU 1101 calculates a difference value between the image quality control parameter of the target divisional block and the predicted image quality control parameter to acquire an image quality control parameter difference value.
- [0030] In step S1050, the CPU 1101 codes the image quality control parameter difference value to generate an image quality control parameter difference value code. The image quality control parameter difference value code is embedded in a bit stream in a structure as illustrated in Fig. 18A.
- [0031] In step S1060, the CPU 1101 codes the image data of the target divisional block. In the present exemplary embodiment, according to the prediction mode, the CPU 1101 generates residual data of the target divisional block and applies orthogonal transform, quantization, and variable-length coding to the residual data to code the residual data and generate variable-length coding coefficient data. In this case, the variable-length coding coefficient data is embedded in a bit stream in the structure as illustrated in Fig. 18A.
- [0032] Fig. 20 is a flowchart illustrating a predicted image quality control parameter acquisition method for each block size determined in step S1030 in Fig. 1. In this flowchart, final image quality control parameters 64, 32, 16, and 8 are used. These parameters are commonly used in processing in the frame, and are to be initialized when decoding is started or for each processing unit such as frame and slice.
- [0033] In step S20001, the CPU 1101 determines the block size of the target block, and performs processing for each block size. Specifically, when the block size is 64x64, 32x32, 16x16, and 8x8 pixels (64x64, 32x32, 16x16, and 8x8 in step S20001, respectively), the processing proceeds to steps S20002 and S20003, steps S20004 and S20005, steps S20006 and S20007, and steps S20008 and S20009, respectively.
- [0034] In step S20002, the final image quality control parameter 64 is set as a predicted image quality control parameter.

- [0035] In step S20003, the CPU 1101 records the image quality control parameter of the target divisional block to the final image quality control parameter 64. Recording a parameter value in this way enables easily referring to the parameter value next time a 64x64 block is processed.
- [0036] Subsequently, when the block size is 32x32, 16x16, 8x8 pixels, the CPU 1101 executes similar processing in steps S20004 and S20005, steps S20006 and S20007, and steps S20008 and S20009, respectively. A predicted image quality control parameter acquired in these steps will be used in step S1040.
- [0037] Fig. 22 illustrates referential relations between image quality control parameters of respective blocks by using arrows. Referring to Fig. 22, for example, when a block B2 is a target divisional block, an image quality control parameter of a last coded block B1 having a similar block size as the target divisional block is set as a predicted image quality control parameter of the block B2. When a block C5 is a target divisional block, an image quality control parameter of a last coded block C4 having a similar block size as the target divisional block is set as a predicted image quality control parameter of the block C5. Thus, in the present exemplary embodiment, image quality control parameters of blocks having different sizes are referred to as predicted image quality control parameters independently for each block size. In many cases, blocks having a similar attribute are coded based on image quality control parameters having identical or close values. For example, there is a tendency that the coding efficiency is improved by coding an image block having many low-frequency components based on a large block size and by coding an image block having many high-frequency components, such as an edge, based on a small block size. Therefore, the block size at the time of block division is also likely to be determined according to this tendency. When controlling image quality control parameters aiming at improving the image quality of a region having a specific image component, image quality control parameters are highly likely to have close values for each block size. Therefore, the technique according to the present exemplary embodiment provides a smaller absolute value of a difference value than a case of calculating a difference value simply by using an image quality control parameter of a last coded block, as a predicted image quality control parameter. This means that the technique provides a smaller amount of codes at the time of image quality control parameter coding.
- [0038] Although, in the present exemplary embodiment, the lattice block size is 64x64 pixels and the minimum divisional block size is 8x8 pixels, the block size is not limited thereto. Further, the coding method according to the present exemplary embodiment may be configured by dedicated hardware such as an orthogonal transform unit 1003, a quantization unit 1004, and a variable-length coding unit 1005, as illustrated in Fig. 11B. Further, the image quality control parameter calculation method according to the

present exemplary embodiment may be configured by dedicated hardware such as a code amount control unit (not illustrated). In this case, the CPU 1101 executes similar processing to the flowcharts in Figs. 1 and 15. However, the CPU 1101 does not execute orthogonal transform processing, quantization processing, variable-length coding processing, and control processing, but instructs the hardware to execute these pieces of processing.

- [0039] An image decoding method according to a second exemplary embodiment of the present invention for decoding coded data generated by the image coding method according to the first exemplary embodiment will be described below with reference to Figs. 6, 11A, 11B, 11C, 16, and 20.
- [0040] Fig. 11A illustrates a hardware configuration for executing a decoding program including the decoding method according to the second exemplary embodiment. Decoding processing is achieved when the CPU 1101 loads the decoding program from the HDD 1104 into the RAM 1102, and executes each step of a flowchart (described below). Input code data is read from the HDD 1104 and a decoded image is output to a display via a video RAM (VRAM) 1105. Although the decoding processing is performed on a block basis, a decoded image for one frame is completed when decoded images of blocks acquired by decoding respective blocks are copied to a suitable position in the RAM 1102 and the decoding processing is completed for all blocks of one frame ends. Then, the generated frame is output to the VRAM 1105.
- [0041] An input bit stream is the bit stream generated in the first exemplary embodiment.
- [0042] Fig. 16 is a flowchart illustrating processing for decoding the input bit stream. Each lattice block is decoded in order of raster scanning.
- [0043] In step S16010, the CPU 1101 decodes the frame header information illustrated in Fig. 17A. In step S16020, the CPU 1101 sets a target lattice block subjected to processing. The target lattice block subjected to the first processing is the lattice block at the top left corner of the screen. Subsequently, each time step S16020 is executed, the CPU 1101 sets a target lattice block in order of raster scanning, for example, in order of the lattice block numbers (B1, B2, ...) illustrated in Fig. 13A.
- [0044] In step S16030, the CPU 1101 decodes the division state of the target lattice block, and the prediction mode for each divisional block. In this step, the region quadtree structure of divisional blocks is restored based on the division flag. Prediction mode information is also decoded for each divisional block, and the attribute information of each divisional block can be acquired.
- [0045] In step S16040, the CPU 1101 sets a target divisional block based on the order of the region quadtree structure restored in step S16030, for example, in order of the divisional block number (b1, b2, ...) illustrated in Fig. 13B. In step S16050, the CPU 1101 decodes image data of the target divisional block. Detailed processing in this step

will be described below with reference to Fig. 6.

- [0046] In step S16060, the CPU 1101 determines whether processing is completed for all divisional blocks in the lattice block. When processing is completed for all divisional blocks (YES in step S16060), the processing proceeds to perform S16070. Otherwise, when processing is not completed for all divisional blocks (NO in step S16060), the processing returns to step S16040.
- [0047] In step S16070, the CPU 1101 determines whether processing is completed for all lattice blocks in the frame. When processing is completed for all lattice blocks (YES in step S16070), the processing exits this flowchart. Otherwise, when processing is not completed for all lattice blocks (NO in step S16070), the processing returns to step S16020 to reexecute steps S16020 to S16070.
- [0048] Fig. 6 is a detailed flowchart illustrating processing executed in step S16050 in Fig. 16. In step S6010, the CPU 1101 acquires attribute information of the target divisional block. Although, in the present exemplary embodiment, the block size is used as attribute information, the attribute information is not limited thereto. In step S6020, the CPU 1101 decodes the image quality control parameter difference value code to generate an image quality control parameter difference value.
- [0049] In step S6030, the CPU 1101 sets a last decoded divisional block having attribute information identical to that of the target divisional block acquired in step S6010, as a predicted image quality control parameter. In the present exemplary embodiment, since the block size is used as attribute information, the CPU 1101 records and refers to an image quality control parameter for each block size.
- [0050] In step S6040, the CPU 1101 adds the image quality control parameter difference value to the predicted image quality control parameter to calculate an image quality control parameter of the target block.
- [0051] In step S6050, the CPU 1101 decodes the image data of the target divisional block. In the decoding processing for a target block according to the present exemplary embodiment, the CPU 1101 applies variable-length decoding, inverse quantization, and inverse orthogonal transform to variable-length coding coefficient data to generate residual data. Further, according to the prediction mode, the CPU 1101 generates a predicted image and adds the residual data to the predicted image to obtain a decoded image of the divisional block. At the time of inverse quantization, the CPU 1101 uses the image quality control parameter calculated in step S6040 as a quantization parameter.
- [0052] Applying the technique according to the second exemplary embodiment enables decoding a bit stream coded by the image coding method according to the first exemplary embodiment.
- [0053] The decoding method according to the second exemplary embodiment may be

configured by dedicated hardware such as an inverse orthogonal transform unit 1304, an inverse quantization unit 1305, a variable-length decoding unit 1306, and a code amount control unit (not illustrated), as illustrated in Fig. 11C. In this case, the CPU 1101 executes similar processing to the flowcharts in Figs. 6 and 16. However, the CPU 1101 does not execute inverse orthogonal transform, inverse quantization, and variable-length decoding processing, but instructs the hardware to execute these pieces of processing.

[0054] An image coding method according to a third exemplary embodiment of the present invention will be described below with reference to Fig. 2. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the first exemplary embodiment.

[0055] Fig. 2 is a detailed flowchart illustrating a divisional block coding method. In this flowchart, step S2021 is added following step S1020 in Fig. 1. In the third exemplary embodiment, the target divisional block is an inter-block.

[0056] In step S2021, the CPU 1101 selects one neighboring block having attribute information closest to that of the target divisional block. Fig. 12B illustrates a relation between the motion vector of the target divisional block and motion vectors of neighboring blocks. Although, in the present exemplary embodiment, neighboring blocks are considered to be blocks to the left, upside, and upper right of the target block as illustrated in Fig. 12B, the combination of blocks is not thereto. Although, in the present exemplary embodiment, selecting a block having attribute information closest to that of the target divisional block is equivalent to selecting a block providing a smallest difference value between the motion vector of the target divisional block and each of motion vectors of neighboring blocks, the processing is not limited thereto. When there is motion vector derivation block identification information for identifying a neighboring block having a similar motion vector as the target divisional block, the information may be used to select a block. Referring to Fig. 12B, for example, the target divisional block has a motion vector V_{Curr} , the left block has a motion vector V_{Left} , the upper block has a motion vector V_{Top} , and the upper right block has a motion vector $V_{TopRight}$. In this case, since the motion vectors V_{Curr} and V_{Top} have a similar vector value, the upper block is selected. Although, in the present exemplary embodiment, when a plurality of blocks has closest or identical attribute information, a spatially closest block is selected, but the processing is not limited thereto.

[0057] In step S2030, the CPU 1101 sets the image quality control parameter of the block selected in step S2021 as a predicted image quality control parameter.

[0058] In many cases, a neighboring block having a similar attribute as the target divisional block is coded based on an image quality control parameter having an identical or close value. For example, blocks having a similar motion vector are highly likely to be

included in a similar object. When performing processing for improving the image quality of a specific object, blocks having a similar motion vector are highly likely to be coded based on image quality control parameters having close values. Therefore, this technique provides a smaller absolute value of a difference value than a case of calculating a difference value simply by using an image quality control parameter of a last coded block, as a predicted image quality control parameter. This means that the technique provides a smaller amount of codes at the time of image quality control parameter coding. Therefore, a moving image having the equivalent image quality can be coded with a less bit rate.

- [0059] An image decoding method according to a fourth exemplary embodiment of the present invention for decoding coded data generated by the image coding method according to the third exemplary embodiment will be described below with reference to Fig. 7. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the second exemplary embodiment. An input bit stream is the bit stream generated in the third exemplary embodiment.
- [0060] Fig. 7 is a detailed flowchart illustrating the divisional block decoding processing in step S16050 in Fig. 16. In this flowchart, step S7021 is added following step S6020 in Fig. 6. In the fourth exemplary embodiment, the target block is an inter-block.
- [0061] In step S7021, the CPU 1101 selects one neighboring block having attribute information closest to that of the target divisional block. Referring to Fig. 12B, similar to the third exemplary embodiment, since the motion vectors $VCurr$ and $VTop$ have a similar vector value, the upper block is selected. In this case, motion vectors may be directly compared to select a block. When there exists information for identifying a neighboring block having a similar motion vector as the target divisional block, the information may be used to select a block. Although, in the present exemplary embodiment, when a plurality of blocks has closest or identical attribute information, a spatially closest block is selected, but the processing is not limited thereto.
- [0062] In step S7030, the CPU 1101 sets the image quality control parameter of the block selected in step S6021 as a predicted image quality control parameter. The predicted image quality control parameter determined here will be used in step S6040. Applying the technique according to the present exemplary embodiment enables decoding a bit stream coded by the image coding method according to the third exemplary embodiment.
- [0063] An image coding method according to a fifth exemplary embodiment of the present invention will be described below with reference to Fig. 3. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the first exemplary embodiment.

- [0064] Fig. 3 is a detailed flowchart illustrating the divisional block coding method illustrated in step S15050 in Fig. 15. In this flowchart, step S3021 is added following step S1020 in Fig. 1.
- [0065] In step S3021, the CPU 1101 selects at least one neighboring block having similar attribute information as the target divisional block. Although, in the present exemplary embodiment, neighboring blocks are considered to be blocks to the left, upside, and upper right of the target block as illustrated in Fig. 12A, the combination of blocks is not limited thereto. In the present exemplary embodiment, the block mode information is used as attribute information. Specifically, when the target divisional block is an intra-block, the CPU 1101 selects a neighboring intra-block. When the target divisional block is an inter-block, the CPU 1101 selects a neighboring inter-block.
- [0066] In step S3030, the CPU 1101 calculates an average of image quality control parameters of at least one selected block, and sets it as a predicted image quality control parameter. Although, in the present exemplary embodiment, an average is used as a predicted image quality control parameter, the predicted image quality control parameter is not limited thereto, and may be a median.
- [0067] Figs. 23A, 23B, 23C, and 23D illustrate example processing according to the fifth exemplary embodiment. Referring to Fig. 23A, since a block to the left of the target divisional block is an inter-block, the left block is selected. Referring to Fig. 23B, since a block to the upside of the target divisional block is an inter-block, the upper block is selected. Referring to Fig. 23C, since blocks to the left and upside of the target divisional block are inter-blocks, an average of the left and upper blocks is selected. Referring to Fig. 23D, since blocks to the left and upside of the target divisional block are not inter-blocks, the left block is selected.
- [0068] In many cases, a neighboring block having a similar attribute as the target divisional block is coded based on an image quality control parameter having an identical or close value. For example, a block at an edge portion of a moving subject is easily coded by intra-coding, and a block on the background or in a subject is easily coded by inter-coding. When performing code amount control for improving the image quality giving priority to an edge portion, it is effective to set an image quality control parameter to intra-coding blocks aiming for high image quality. Specifically, image quality control parameters of blocks having similar block mode information tend to have close values. Therefore, the technique according to the fifth exemplary embodiment provides a smaller absolute value of a difference value than a case of calculating a difference value simply by using an image quality control parameter of a last coded block, as a predicted image quality control parameter. This means that the technique provides a smaller amount of codes at the time of image quality control parameter coding.

- [0069] An image decoding method according to a sixth exemplary embodiment of the present invention for decoding coded data generated by the image coding method according to the fifth exemplary embodiment will be described below with reference to Fig. 8. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the second exemplary embodiment. An input bit stream is the bit stream generated in the fifth exemplary embodiment.
- [0070] Fig. 8 is a detailed flowchart illustrating the divisional block decoding processing in step S16050 in Fig. 16. In this flowchart, step S8021 is added following step S6020 in Fig. 6.
- [0071] In step S8021, the CPU 1101 selects at least one neighboring block having similar attribute information as the target divisional block. Although, in the sixth exemplary embodiment, neighboring blocks are considered to be blocks to the left, upside, and upper right of the target block as illustrated in Fig. 12A, the combination of blocks is not limited thereto. Specifically, when the target divisional block is an intra-block, the CPU 1101 selects a neighboring intra-block. When the target divisional block is an inter-block, the CPU 1101 selects a neighboring inter-block.
- [0072] In step S8030, the CPU 1101 calculates an average of image quality control parameters of at least one selected block, and sets it as a predicted image quality control parameter. Although, in the sixth exemplary embodiment, an average is used as a predicted image quality control parameter, the predicted image quality control parameter is not limited thereto, and may be a median. The predicted image quality control parameter determined here will be used in step S6040.
- [0073] Applying the technique according to the sixth exemplary embodiment enables decoding a bit stream coded by the image coding method according to the fifth exemplary embodiment.
- [0074] An image coding method according to a seventh exemplary embodiment of the present invention will be described below with reference in Fig. 4. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the first exemplary embodiment.
- [0075] Fig. 4 is a flowchart illustrating a divisional block coding method illustrated in step S15050 in Fig. 15. In this flowchart, steps S4021, S4022, and S4023 are added following step S1020 in Fig. 1.
- [0076] In step S4021, the CPU 1101 selects a predicted image quality control parameter determination method from a plurality of candidates. Although, in the present exemplary embodiment, the CPU 1101 selects a predicted image quality control parameter determination method out of the following four different methods ((0), (1), (2), and (3)), the combination is not limited thereto, and may be other determination methods or a part

of the four methods. (0) Set an image quality control parameter of a divisional block spatially existing to the left of the target divisional block, as a predicted image quality control parameter. (1) Set an average of image quality control parameters of neighboring divisional blocks having similar attribute information as the target divisional block, as a predicted image quality control parameter. (2) Set a median of image quality control parameters of neighboring divisional blocks having similar attribute information as the target divisional block, as a predicted image quality control parameter. (3) Set an image quality control parameter of a last coded divisional block having similar attribute information as the target divisional block, as a predicted image quality control parameter. As concrete processing in step S4021, the CPU 1101 attempts the above-mentioned predicted image quality control parameter determination methods, and selects a predicted image quality control parameter determination method which provides a smallest difference value between the image quality control parameter of the target divisional block and the predicted image quality control parameter.

[0077] In step S4022, the CPU 1101 determines an identification value for identifying the predicted image quality control parameter determination method. In the present exemplary embodiment, the CPU 1101 sets an index value corresponding to the predicted image quality control parameter determination method determined in step S4021, as an identification value.

[0078] In step S4023, the CPU 1101 codes the above-mentioned identification value to generate an identification value code.

[0079] In step S4030, the CPU 1101 acquires a predicted image quality control parameter calculated by the predicted image quality control parameter determination method selected in step S4021.

[0080] Fig. 18B illustrates a structure of a bit stream of a lattice block according to the seventh exemplary embodiment. Transmitting the bit stream to the decoding unit enables identifying the predicted image quality control parameter determination method also on side of the decoding unit.

[0081] According to the seventh exemplary embodiment, an optimum predicted image quality control parameter determination method can be selected. In the present exemplary embodiment, an optimum predicted image quality control parameter determination method is selected for each divisional block, and then the identification value is coded. However, the processing is not limited thereto. It is also possible to select an optimum predicted image quality control parameter determination method in unit of lattice block, slice, frame, or sequence, and embed the identification value in the header of the above-mentioned coding unit. For example, the identification value is embedded in the frame header in Fig. 17A.

- [0082] An image decoding method according to an eighth exemplary embodiment of the present invention for decoding coded data generated by the image coding method according to the seventh exemplary embodiment will be described below with reference in Fig. 9. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the second exemplary embodiment. An input bit stream is the bit stream generated in the seventh exemplary embodiment.
- [0083] Fig. 9 is a detailed flowchart illustrating the divisional block decoding processing in step S16050 in Fig. 16. In this flowchart, step S9021 is added following step S6020 in Fig. 2. In the eighth exemplary embodiment, the CPU 1101 decodes the bit stream generated in the seventh exemplary embodiment illustrated in Fig. 18B.
- [0084] In step S6010, the CPU 1101 acquires attribute information of the target divisional block. Although, in the eighth exemplary embodiment, the block mode information for identifying intra- and inter-prediction is used as attribute information, the attribute information is not limited thereto. For example, the block size, motion vector, or intra-prediction method in intra-prediction may be used alone or in combination.
- [0085] In step S9021, the CPU 1101 decodes the identification value code for identifying the predicted image quality control parameter determination method to generate an identification value. In the eighth exemplary embodiment, although the identification value corresponds to (0), (1), (2), and (3) described in the seventh exemplary embodiment, the identification value is not limited thereto.
- [0086] In step S9030, the CPU 1101 determines a predicted image quality control parameter by using the predicted image quality control parameter determination method identified by the above-mentioned identification value. The predicted image quality control parameter determined here will be used in step S6040.
- [0087] The decoding method according to the eighth exemplary embodiment enables decoding a bit stream coded by the image coding method according to the seventh exemplary embodiment.
- [0088] An image coding method according to a ninth exemplary embodiment of the present invention will be described below with reference in Fig. 5. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the first exemplary embodiment.
- [0089] Fig. 5 is a detailed flowchart illustrating the divisional block coding processing in step S15050 in Fig. 15. In this flowchart, step S5021 is added following step S1020 in Fig. 1, and steps S5041 and S5042 are added following step S5040.
- [0090] In step S5021, the CPU 1101 selects a plurality of neighboring blocks having similar attribute information as the target divisional block. Although, in the present exemplary embodiment, blocks to the left, upside, and upper right of the target block are selected

as illustrated in Fig. 12A, the combination is not limited thereto. Although, in the ninth exemplary embodiment, all of the left, upper, and upper right blocks are selected when they have different modes, the combination is not limited thereto.

- [0091] In step S5030, the CPU 1101 calculates a difference value between the image quality control parameter of the target divisional block and each of image quality control parameters of the selected blocks, and selects a block providing a smallest difference value. In step S5040, the CPU 1101 acquires a difference value calculated by using the image quality control parameter of the block selected in step S5030. This difference value will be coded in step S1050.
- [0092] In step S5041, the CPU 1101 calculates an identification value for identifying a block subjected to image quality control parameter prediction out of the plurality of blocks. For example, this identification value can be calculated with reference to a table in Fig. 14. Referring to Fig. 14, "SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE" indicates whether divisional blocks existing to the left, upside, and upper right of the target divisional block are selected or not in step S5021. Identification codes in this table are assigned according to "SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE", and the identification codes corresponding to the left, upper, and upper right divisional blocks selected in step S5021 are used as identification values. In the table, "-" indicates that no identification code is assigned since no block is selected, and "x" indicates that a block can be uniquely identified even without coding the identification value.
- [0093] In step S5042, the CPU 1101 codes the above-mentioned identification value to generate an identification value code. In the ninth exemplary embodiment, the CPU 1101 codes the above-mentioned identification value by using the coding method described in column "CODING METHOD" according to "SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE" in Fig. 14. In cases 1 and 8 in Fig. 14, the CPU 1101 generates variable-length Golomb codes and embeds them in a bit stream. In cases 2, 3, and 5 in Fig. 14, the CPU 1101 embeds values in a bit stream as 1-bit fixed length data. In cases 4, 6, and 7 in Fig. 14, the identification value code is not embedded in a bit stream since blocks can be uniquely identified. However, the coding method is not limited thereto.
- [0094] According to the ninth exemplary embodiment, it is possible to select for each divisional block a neighboring divisional block having a predicted image quality control parameter that minimizes the amount of image quality control parameter codes of the target divisional block.
- [0095] An image decoding method according to a tenth exemplary embodiment of the present invention for decoding coded data generated by the image coding method according to the ninth exemplary embodiment will be described below with reference

in Fig. 10. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the second exemplary embodiment. An input bit stream is the bit stream generated in the ninth exemplary embodiment.

[0096] Fig. 10 is a detailed flowchart illustrating the divisional block coding processing in step S16050 in Fig. 16. In this flowchart, steps S10021, S10022, and S10023 are added following step S6020 in Fig. 6.

[0097] In step S10021, the CPU 1101 selects a plurality of neighboring divisional blocks having similar attribute information as the target divisional block. Although, in the present exemplary embodiment, blocks to the left, upside, and upper right of the target block are selected as illustrated in Fig. 12A, the combination is not limited thereto as long as the attribute information is similar as the coding method. Although, in the ninth exemplary embodiment, all of the left, upper, and upper right blocks are selected when all of them have different modes, the combination is not limited thereto as long as the attribute information is similar to the coding method.

[0098] In step S10022, the CPU 1101 decodes the identification value based on the identification value code. Although, in the tenth exemplary embodiment, the CPU 1101 decodes the above-mentioned identification value by using the decoding method corresponding to the coding method described in column "CODING METHOD" according to "SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE" in Fig. 14, the processing is not limited thereto. In cases 2, 3, and 5 in Fig. 14, since a block can be uniquely identified and the identification value code is not embedded in a bit stream, the identification value code is not decoded.

[0099] In step S10023, the CPU 1101 selects a divisional block used for predicted image quality control parameter calculation based on the identification value. Specifically, in the present exemplary embodiment, the CPU 1101 identifies the left, upper, and upper right blocks based on "SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE" and the identification value in Fig. 14. Also in cases where the identification value code is not decoded in step S10022, the CPU 1101 can select one prediction block based on "SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE" in Fig. 14. The block selected here will be used in step S6030.

[0100] The decoding method according to the tenth exemplary embodiment enables decoding a bit stream coded by the image coding method according to the ninth exemplary embodiment.

[0101] An image coding method according to a eleventh exemplary embodiment of the present invention will be described below with reference in Fig. 21. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the first exemplary embodiment.

- [0102] In the present exemplary embodiment, the target divisional block is an intra-block. Although the prediction method for intra-prediction is any of DC prediction, horizontal prediction, vertical prediction, and planar prediction similar to the 16x16 intra-prediction block in H.264, the prediction method is not limited thereto.
- [0103] Fig. 21 is a flowchart illustrating processing for determining a predicted image quality control parameter based on the intra-prediction method in step S1030 in Fig. 1.
- [0104] In step S21001, the CPU determines the intra-prediction method of the target divisional block. When the intra-prediction method is horizontal direction prediction (HORIZONTAL DIRECTION PREDICTION in step S21001), the processing proceeds to step S21002. When the intra-prediction method is DC prediction or planar prediction (DC PREDICTION or PLANAR PREDICTION in step S21001), the processing proceeds to step S21003. When the intra-prediction method is vertical direction prediction (VERTICAL DIRECTION PREDICTION in step S21001), the processing proceeds to step S21004.
- [0105] In step S21002, the CPU 1101 sets the image quality control parameter of the divisional block existing to the left of the target divisional block, as a predicted image quality control parameter.
- [0106] In step S21003, the CPU 1101 sets an average of image quality control parameters of the divisional blocks existing to the left and upside of the target divisional block, as a predicted image quality control parameter.
- [0107] In step S21004, the CPU 1101 sets the image quality control parameter of the divisional block existing to the upside of the target divisional block, as a predicted image quality control parameter.
- [0108] However, the combination of the intra-prediction method and divisional blocks used for predicted image quality control parameter calculation is not limited thereto. For example, when the intra-prediction method has 34 directions as illustrates in Fig. 25, the CPU 1101 calculates predicted image quality control parameters as illustrated in Fig. 24A. In this case, when an index for indicating the prediction direction is 18, 10, or 19, for example, the CPU 1101 calculates an average of image quality control parameters of the divisional blocks existing to the left and upside of the target divisional block (top_left and top in Fig. 24B), as a predicted image quality control parameter of the target divisional block.
- [0109] Blocks existing in the prediction direction of intra-prediction are highly likely to be included in a similar object. When performing processing for improving the image quality of a specific object, these blocks are highly likely to be coded based on image quality control parameters having similar values. Therefore, the technique according to the eleventh exemplary embodiment provides a smaller absolute value of a difference value than a case of calculating a difference value simply by using an image quality

control parameter of a last coded block, as a predicted image quality control parameter. This means that the technique provides a smaller amount of codes at the time of image quality control parameter coding.

- [0110] An image decoding method for decoding coded data generated by the image coding method according to the eleventh exemplary embodiment will be described below with reference in Fig. 21. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the second exemplary embodiment. An input bit stream is the bit stream generated in the eleventh exemplary embodiment.
- [0111] In a twelfth exemplary embodiment of the present invention, the target divisional block is an intra-block. Although the prediction method for intra-prediction is any of DC prediction, horizontal prediction, vertical prediction, and planar prediction similar to the 16x16 intra-prediction block in H.264, the prediction method is not limited thereto.
- [0112] In step S6030 in Fig. 6, the CPU 1101 determines a predicted image quality control parameter based on the intra-prediction method following the flowchart in Fig. 21.
- [0113] In step S21001, the CPU determines the intra-prediction method of the target divisional block. When the intra-prediction method is horizontal direction prediction (HORIZONTAL DIRECTION PREDICTION in step S21001), the processing proceeds to step S21002. When the intra-prediction method is DC prediction or planar prediction (DC PREDICTION or PLANAR PREDICTION in step S21001), the processing proceeds to step S21003. When the intra-prediction method is vertical direction prediction (VERTICAL DIRECTION PREDICTION in step S21001), the processing proceeds to step S21004.
- [0114] In step S21002, the CPU 1101 sets the image quality control parameter of the divisional block existing to the left of the target divisional block, as a predicted image quality control parameter.
- [0115] In step S21003, the CPU 1101 sets an average of image quality control parameters of the divisional blocks existing to the left and upside of the target divisional block, as a predicted image quality control parameter.
- [0116] In step S21004, the CPU 1101 sets the image quality control parameter of the divisional block existing to the upside of the target divisional block, as a predicted image quality control parameter.
- [0117] However, the combination of the intra-prediction method and the divisional blocks used for predicted image quality control parameter calculation is not limited thereto as long as the attribute information is similar to the coding method.
- [0118] Applying the technique according to the twelfth exemplary embodiment enables decoding a bit stream coded by the image coding method according to the eleventh

exemplary embodiment.

- [0119] An image coding method according to a thirteenth exemplary embodiment of the present invention will be described below with reference in Fig. 2. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the first and third exemplary embodiments.
- [0120] In step S2021, the CPU 1101 selects a neighboring divisional block having similar attribute information as the target divisional block. In the thirteenth exemplary embodiment, a motion vector is used as attribute information. In the example in Fig. 12B, since the motion vectors V_{Curr} and V_{Top} have a similar vector value, the upper block is selected.
- [0121] In step S2030, the CPU sets the image quality control parameter of the block selected in step S2021 as a predicted image quality control parameter.
- [0122] In the thirteenth exemplary embodiment, in the case of a similar difference value between the motion vector of the target divisional block and each of motion vectors of neighboring blocks (calculated in step S2021), the image quality control parameter difference value code is not embedded in a bit stream. When the image quality control parameter difference value code is not embedded in a bit stream, the CPU 1101 skips steps S1040 and S1050 not to embed the image quality control parameter difference value code. In image block coding in step S1060, the CPU 1101 does not use the image quality control parameter calculated in step S1020. Instead, the CPU 1101 uses the predicted image quality control parameter determined in step S2030 as a parameter for image quality control.
- [0123] It is useful to code a neighboring block having a similar attribute as the target divisional block, based on a similar image quality control parameter. For example, blocks having a similar motion vector as the target divisional block are highly likely to be included in a similar object. In this case, coding these blocks based on the similar image quality control parameter prevents an image quality difference from occurring in the similar object. By not embedding an image quality control parameter difference value in a bit stream, a moving image having the equivalent image quality can be coded with a less bit rate.
- [0124] It is also possible to insert into a bit stream a code for identifying the existence of a neighboring block having a similar motion vector as the target divisional block and the position of the relevant neighboring block. When there exists a neighboring block having a similar motion vector as the target divisional block, the above-mentioned processing without embedding an image quality control parameter difference value code is performed. In this case, a motion vector is not embedded in the target divisional block. Applying the present exemplary embodiment further enables reducing the amount of codes at the time of image quality control parameter coding.

- [0125] An image decoding method according to a fourteenth exemplary embodiment of the present invention for decoding a bit stream coded by the image coding method according to the first and thirteenth exemplary embodiments will be described below with reference in Fig. 7. Unless otherwise described, the configuration and processing of each step in the present exemplary embodiment are similar to those of the second and fourth exemplary embodiments.
- [0126] In step S7021, the CPU 1101 selects a neighboring divisional block having similar attribute information as the target divisional block. In the fourteenth exemplary embodiment, a motion vector is used as attribute information. In the example in Fig. 12B, since motion vectors V_{Curr} and V_{Top} have a similar vector value, the upper block is selected.
- [0127] In step S7030, the CPU 1101 sets the image quality control parameter of the block selected in step S7021 as a predicted image quality control parameter. The predicted image quality control parameter determined here will be used in step S6040. In the present exemplary embodiment, in the case of a similar difference value between the motion vector of the target divisional block and each of motion vectors of neighboring blocks (calculated in step S7021), the image quality control parameter difference value code is not decoded. When the image quality control parameter difference value code is not decoded, the CPU 1101 skips steps S6020 and S6040 and, in image block decoding in step S7040, uses the predicted image quality control parameter determined in step S7030 as a parameter for image quality control.
- [0128] Applying the technique according to the fourteenth exemplary embodiment enables decoding a bit stream coded by the image coding method according to the third exemplary embodiment.
- [0129] Also when a bit stream includes a code for identifying the existence of a neighboring block having a similar motion vector as the target divisional block and the position of the relevant neighboring block, the above-mentioned image quality control parameter difference value code may not be decoded. In this case, the predicted image quality control parameter is the image quality control parameter of a block having a similar motion vector as the target divisional block, identified by the above-mentioned code. It is necessary that these pieces of processing correspond to the processing of the coding method.
- [0130] Although, in the above-described exemplary embodiments, coded data is recorded on a HDD 903, the recording form of coded data is not limited thereto. For example, coded data may be recorded on a recording medium other than a hard disk drive. Further, coded data may be transmitted to a communication line via a communication interface 905, or recorded on a portable medium in a connected external recording device.

- [0131] Although, in the above-described exemplary embodiments, software is recorded on a hard disk drive, the recording form of software is not limited thereto. For example, the software may be recorded on a hardware unit such as a ROM or a portable medium such as a memory card and a disk.
- [0132] While the present invention has been described with reference to exemplary embodiments, it is to be understood that the invention is not limited to the disclosed exemplary embodiments. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all modifications, equivalent structures, and functions.
- [0133] This application claims priority from Japanese Patent Application No. 2011-051269 filed March 9, 2011, which is hereby incorporated by reference herein in its entirety.

Claims

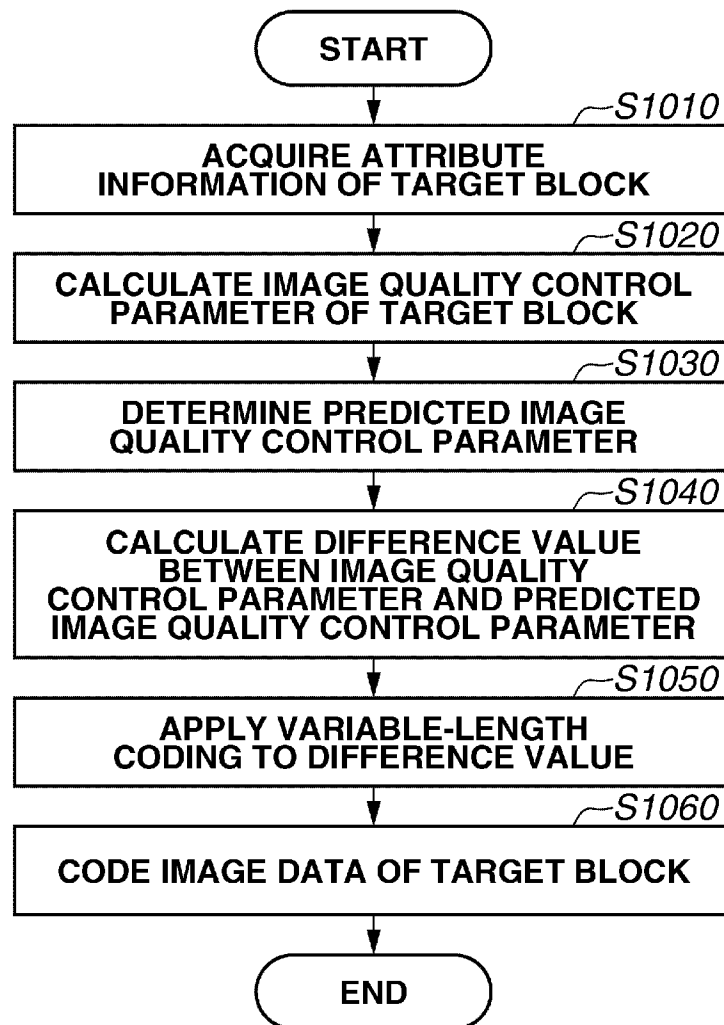
- [Claim 1] An image coding method for dividing an input image into a plurality of divisional blocks having different sizes, and coding the image for each of the divisional blocks, the image coding method comprising:
acquiring attribute information of a target block;
setting a control parameter for controlling an image quality of the target block;
determining a prediction control parameter based on the attribute information;
calculating a difference value between the control parameter and the prediction control parameter; and
coding the calculated difference value.
- [Claim 2] An image decoding method for decoding input coded data, the image decoding method comprising:
decoding, from the input coded data, coded attribute information and a coded difference value of control parameters;
determining a prediction control parameter based on the decoded attribute information; and
calculating a control parameter of a target block based on the prediction control parameter and the decoded difference value.
- [Claim 3] An image coding apparatus configured to divide an input image into a plurality of divisional blocks having different sizes, and to code the image for each of the divisional blocks, the image coding apparatus comprising:
acquisition means configured to acquire attribute information of a target block;
setting means configured to set a control parameter for controlling an image quality of the target block;
determination means configured to determine a prediction control parameter based on the attribute information;
calculation means configured to calculate a difference value between the control parameter set by the setting means and the prediction control parameter; and
coding means configured to code the calculated difference value.
- [Claim 4] An image decoding apparatus configured to decode input coded data, the image decoding apparatus comprising:
decoding means configured to decode, from the input coded data,

coded attribute information and a coded difference value of control parameters;

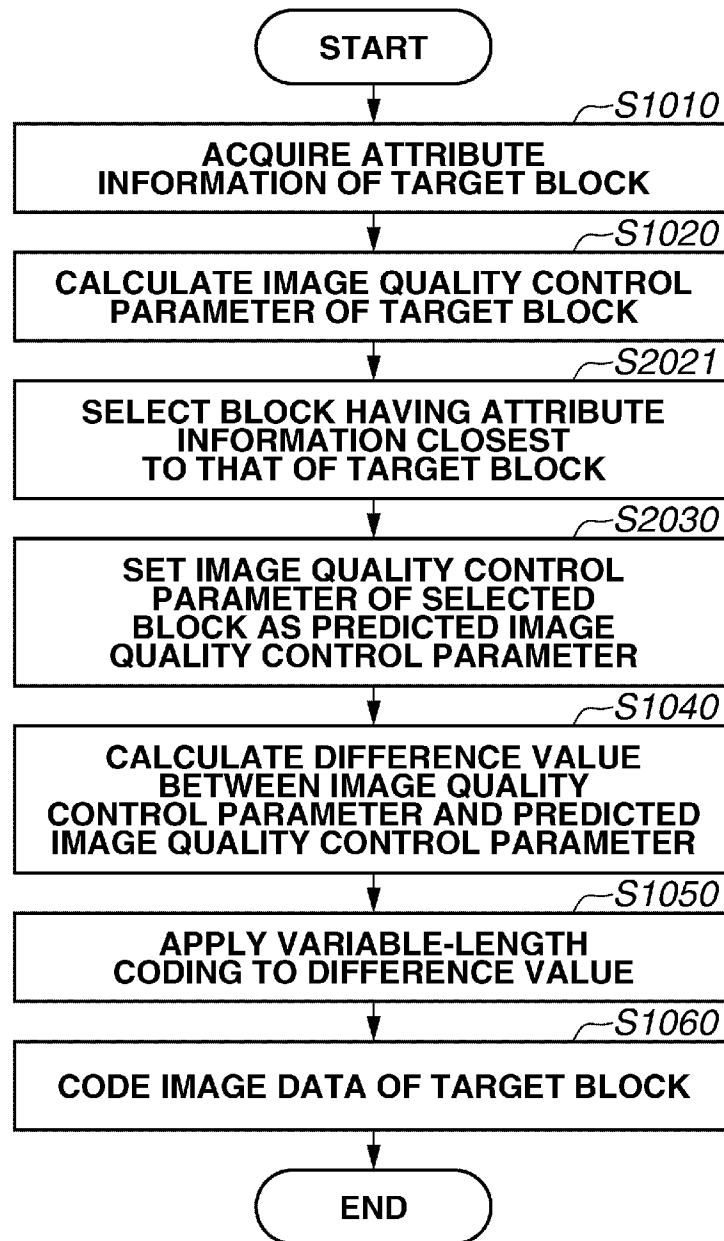
determination means configured to determine a prediction control parameter based on the decoded attribute information; and

calculation means configured to calculate a control parameter of a target block based on the prediction control parameter and the decoded difference value.

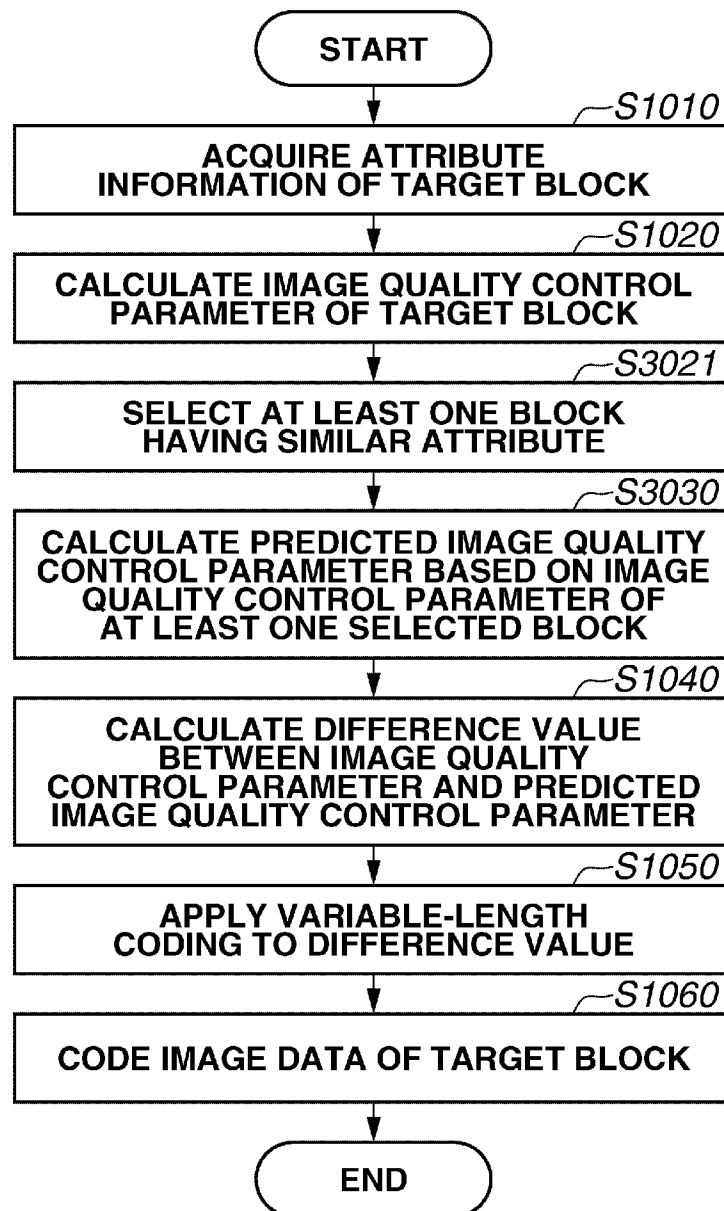
[Fig. 1]



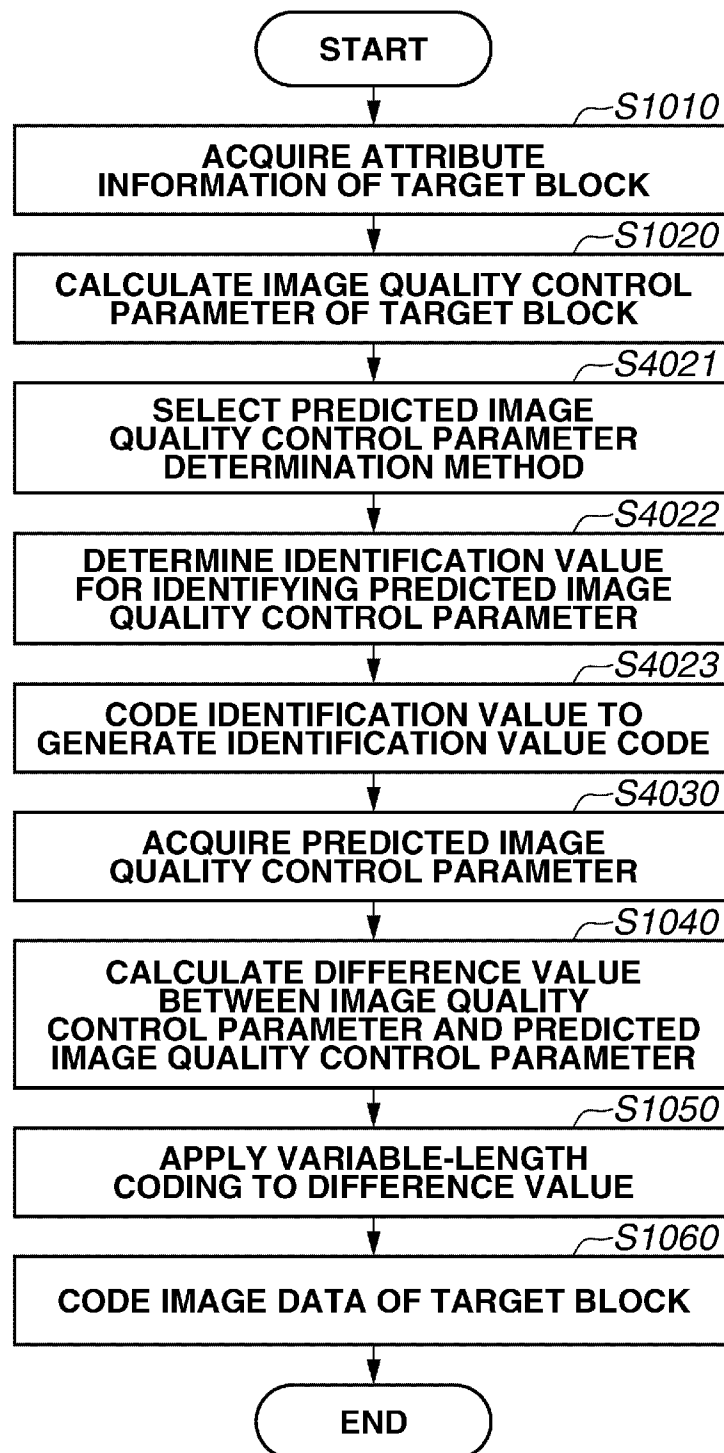
[Fig. 2]



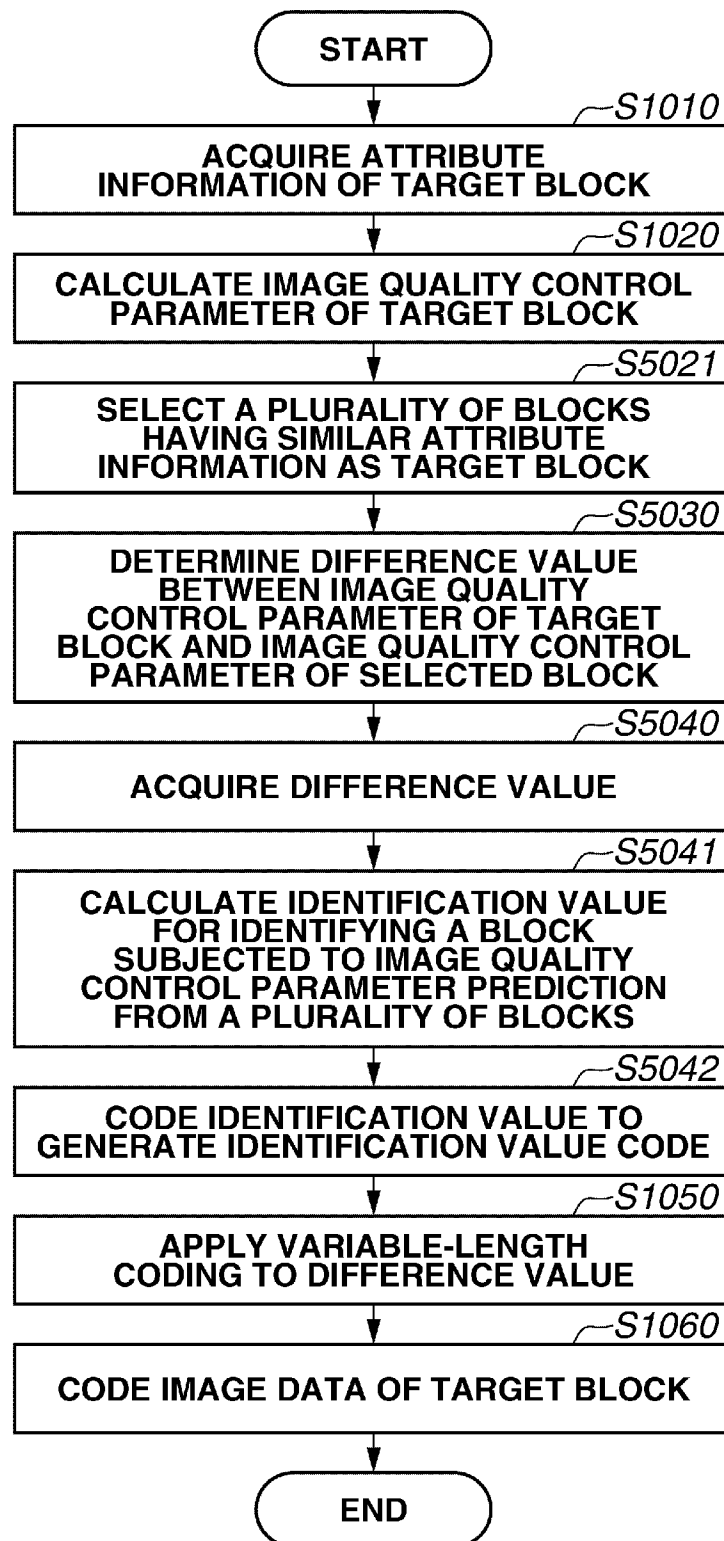
[Fig. 3]



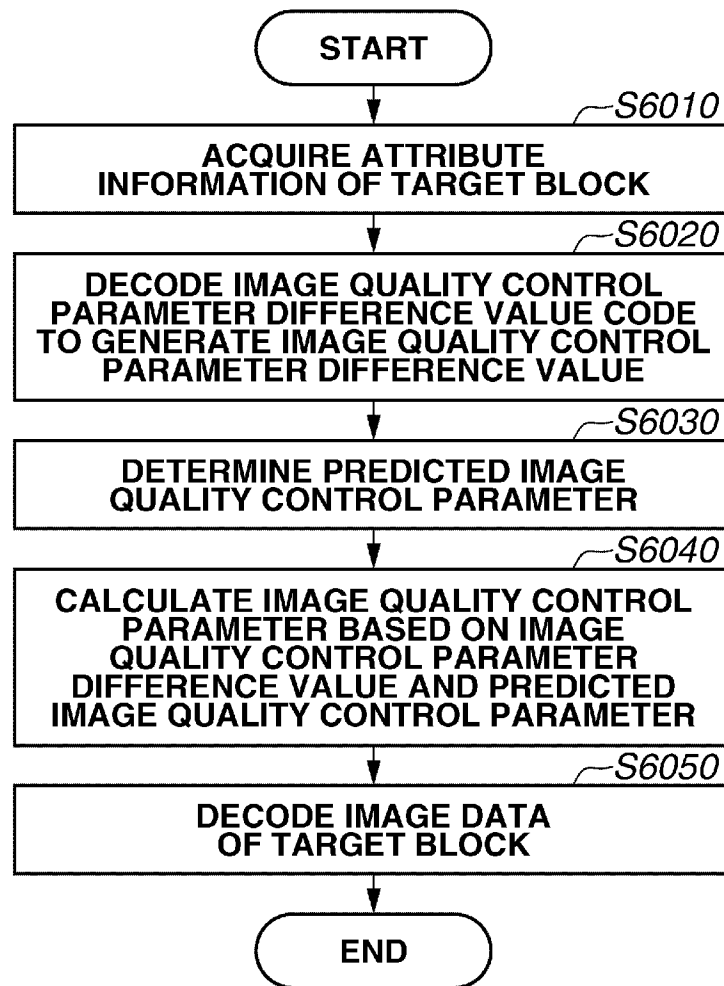
[Fig. 4]



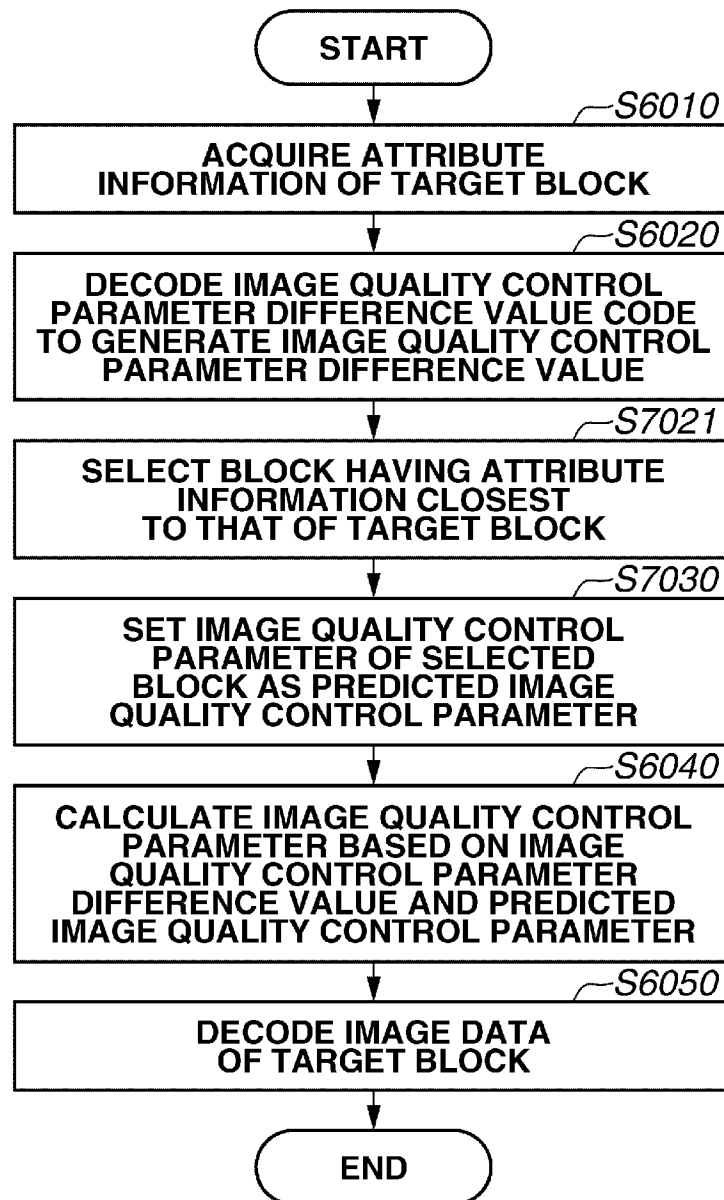
[Fig. 5]



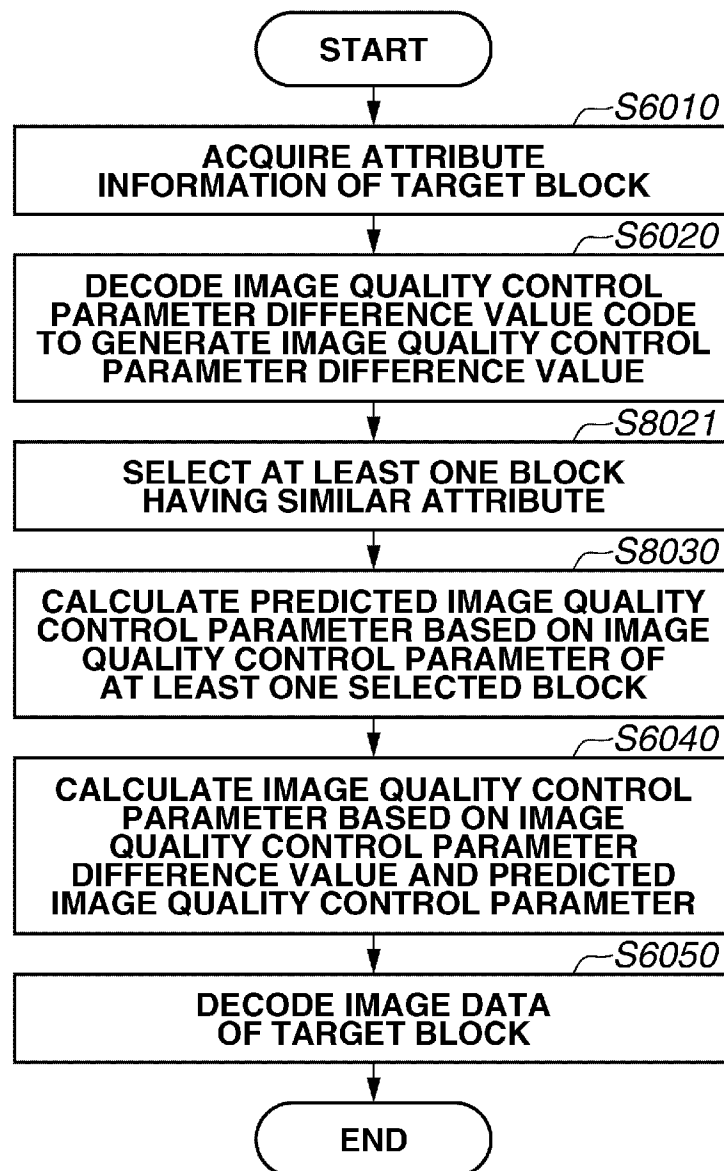
[Fig. 6]



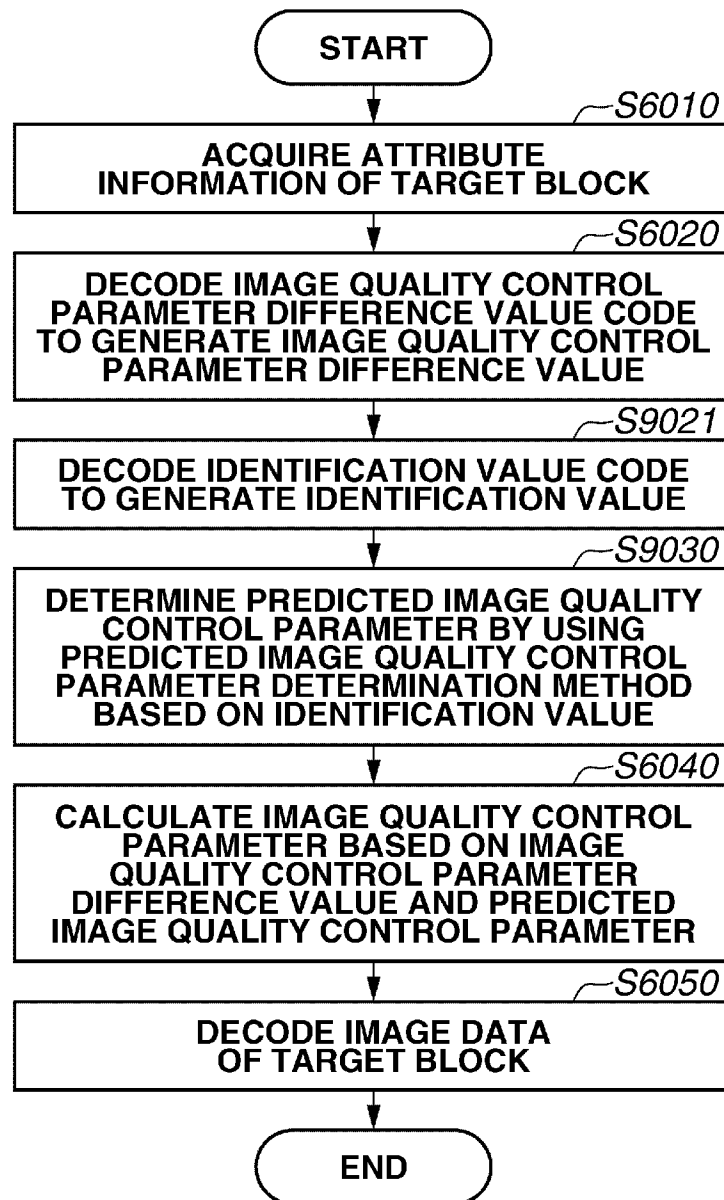
[Fig. 7]



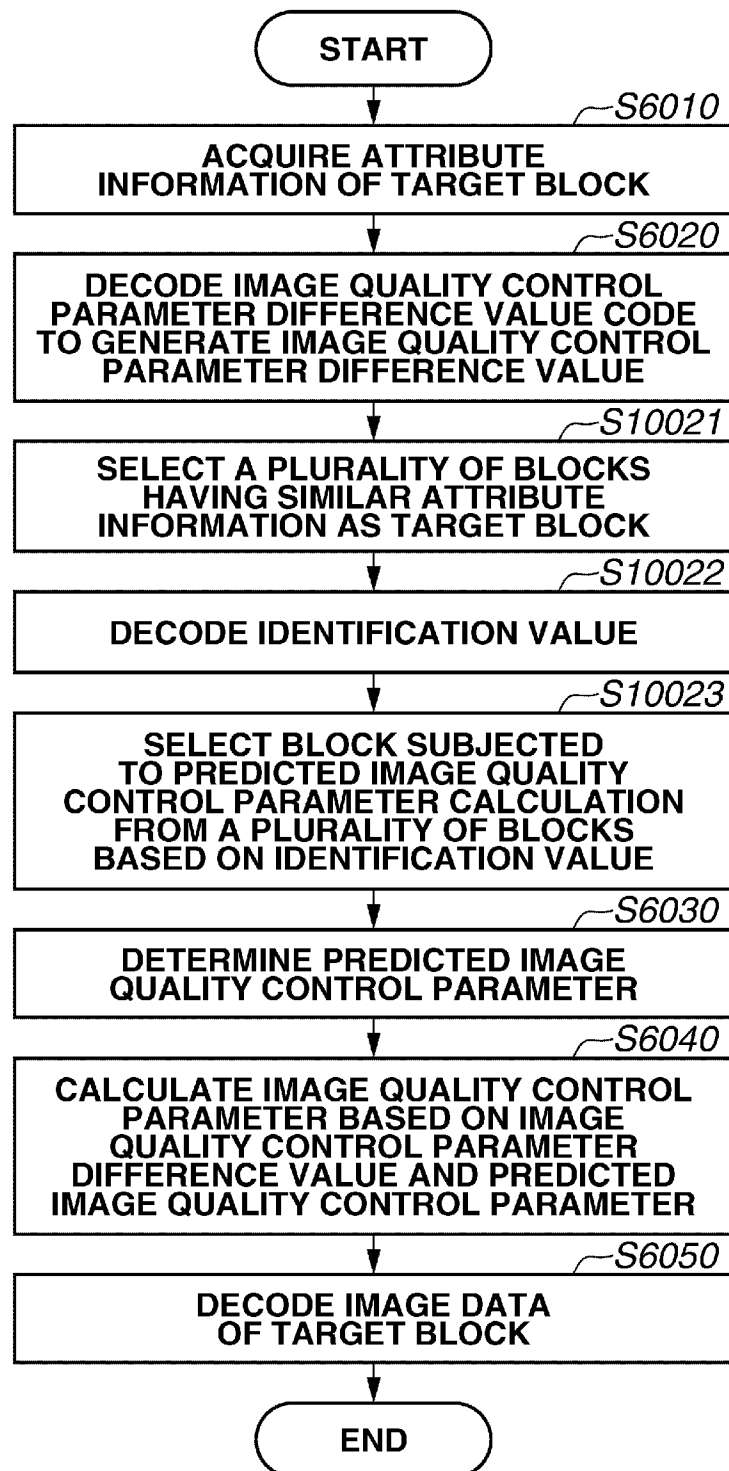
[Fig. 8]



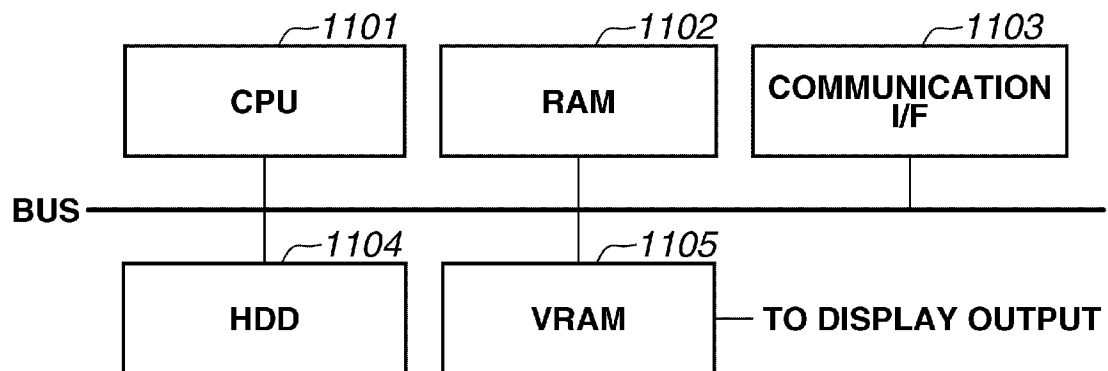
[Fig. 9]



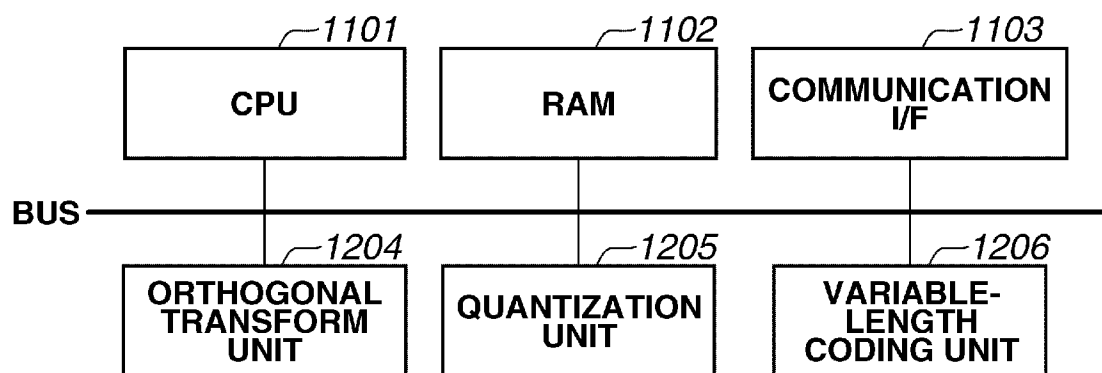
[Fig. 10]



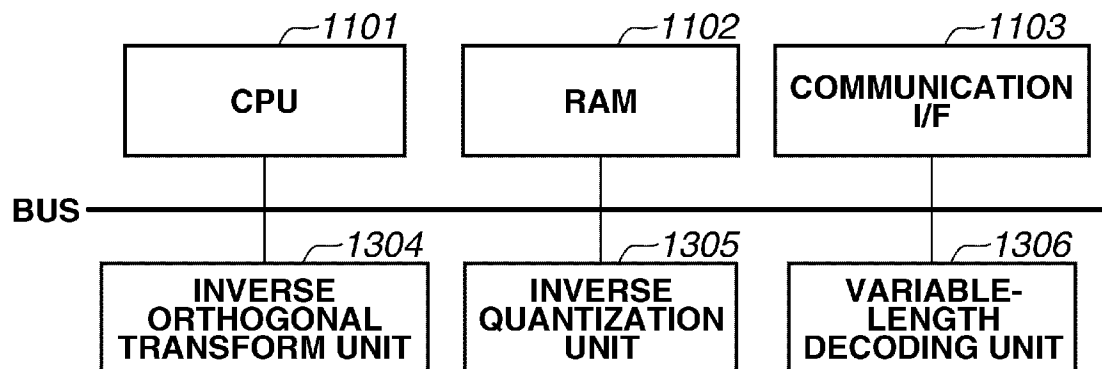
[Fig. 11A]



[Fig. 11B]



[Fig. 11C]



[Fig. 12A]

	UPPER	UPPER RIGHT	
LEFT	TARGET		

[Fig. 12B]

	VTop	VTopRight	
VCurr	UPPER	UPPER RIGHT	
VLeft	LEFT	TARGET	

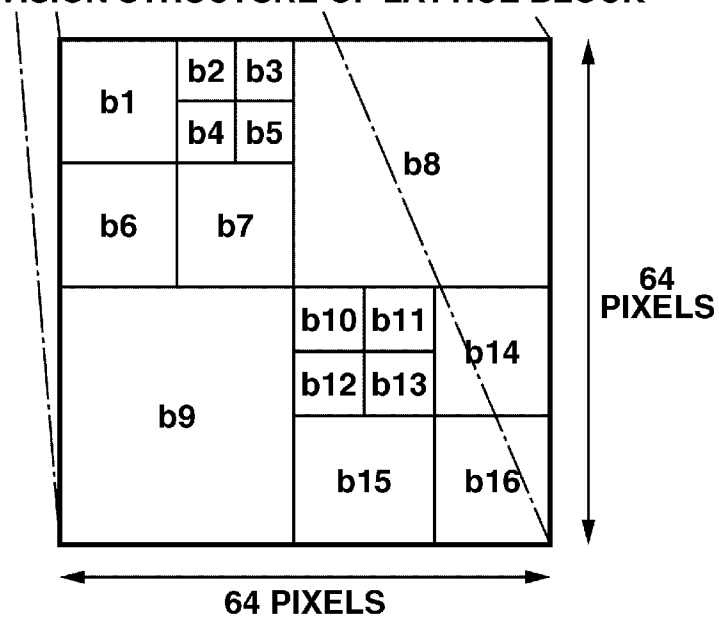
[Fig. 13A]

DIVISION STRUCTURE OF FRAME

B1	B2	B3	B4	B6	B7	B8	B9	B10	B11	B12	B13
B14	B15	B16	B17	B18	B19	B20	B21	B22	B23	B24	B25
B26	B27	B28	B29	B30	B31	B32	B33	B34	B35	B36	B37
B38	B39	B40	B41	B42	B43	B44	B45	B46	B47	B48	B49
B50	B51	B52	B53	B54	B55	B56	B57	B58	B59	B60	B61
B62	B63	B64	B65	B66	B67	B68	B69	B70	B71	B72	B73
B74	B75	B76	B77	B78	B79	B80	B81	B82	B83	B84	B85

[Fig. 13B]

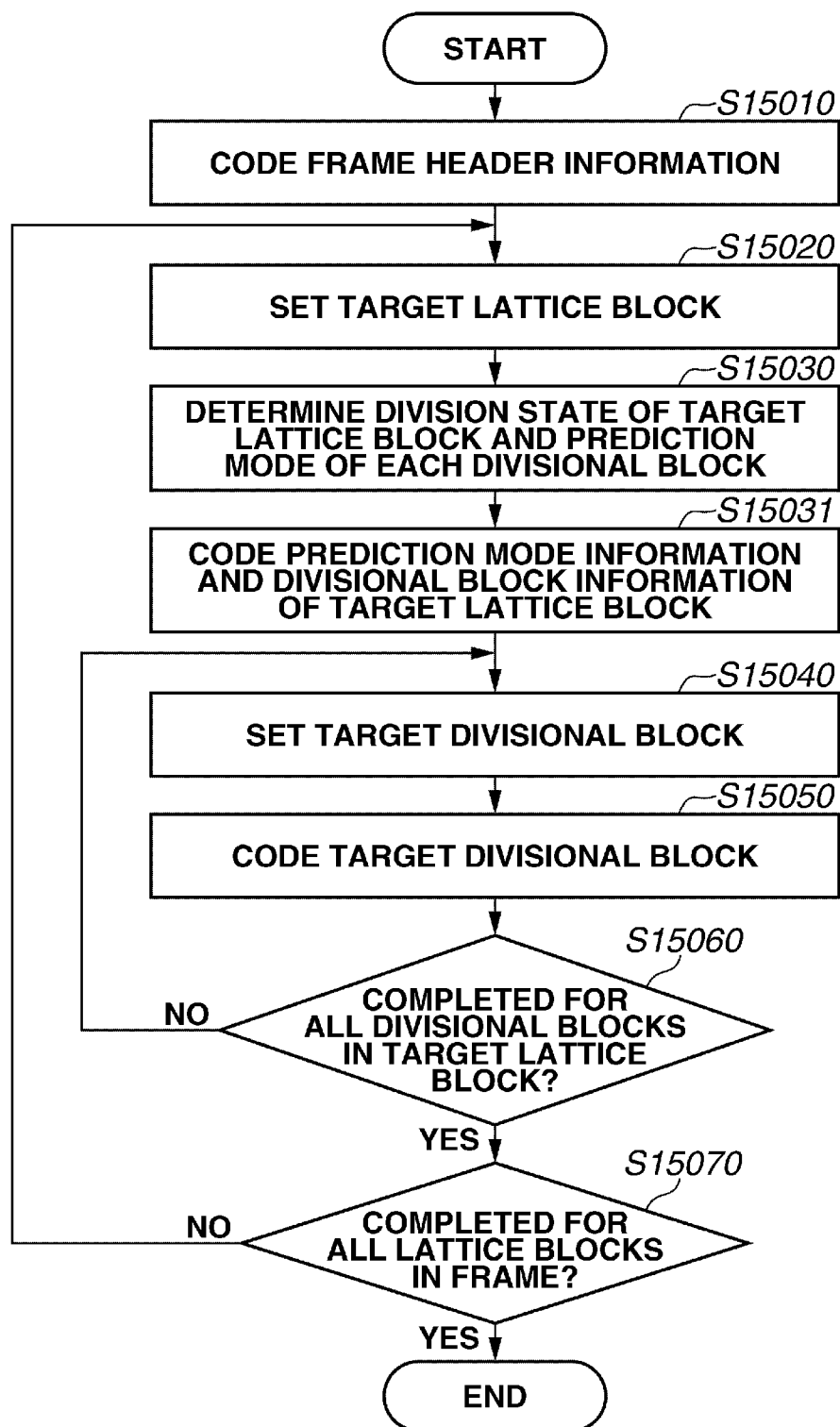
DIVISION STRUCTURE OF LATTICE BLOCK



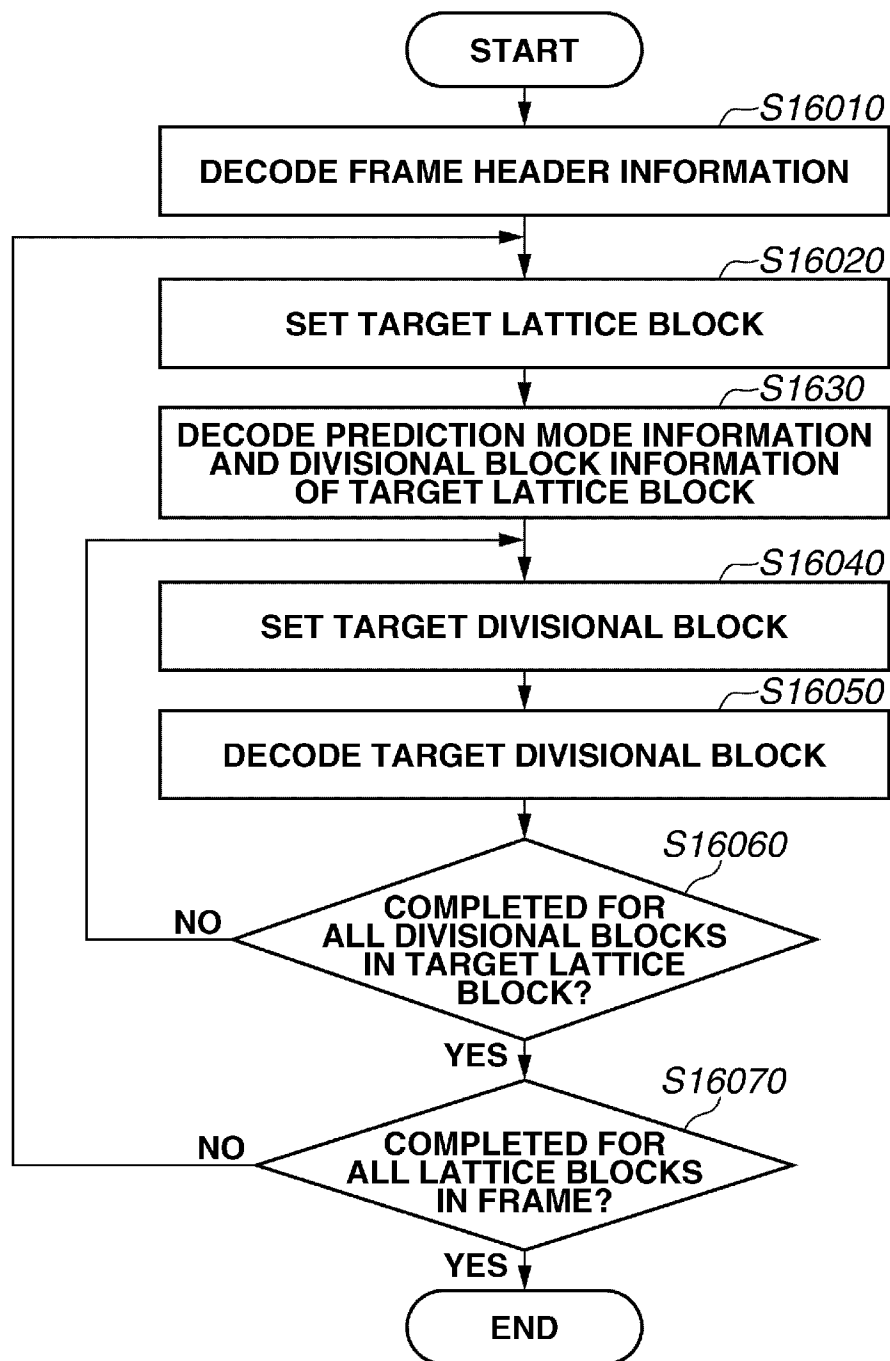
[Fig. 14]

CASE	SELECTION OF BLOCK HAVING SIMILAR ATTRIBUTE			IDENTIFICATION VALUE			CODING METHOD
	LEFT	UPPER	UPPER RIGHT	LEFT	UPPER	UPPER RIGHT	
1	SELECTED	SELECTED	SELECTED	0	1	2	Golomb CODING
2	SELECTED	SELECTED	NOT SELECTED	0	1	—	1-BIT DATA
3	SELECTED	NOT SELECTED	SELECTED	0	—	1	1-BIT DATA
4	SELECTED	NOT SELECTED	NOT SELECTED	×	—	—	NOT SELECTED
5	NOT SELECTED	SELECTED	SELECTED	—	0	1	1-BIT DATA
6	NOT SELECTED	SELECTED	NOT SELECTED	—	×	—	NOT SELECTED
7	NOT SELECTED	NOT SELECTED	SELECTED	—	—	×	NOT SELECTED
8	NOT SELECTED	NOT SELECTED	NOT SELECTED	0	1	2	Golomb CODING

[Fig. 15]

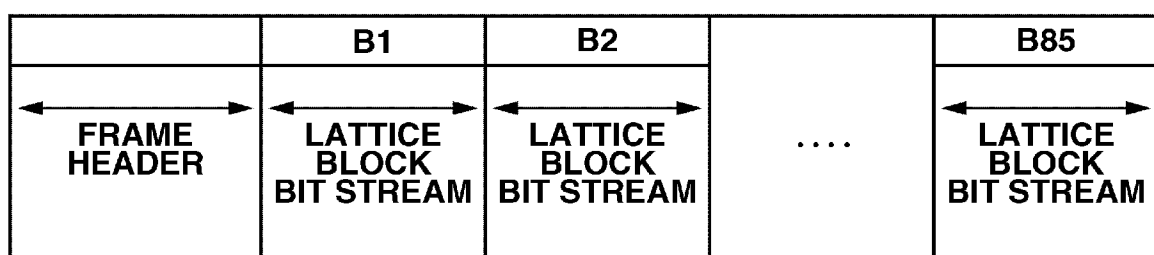


[Fig. 16]



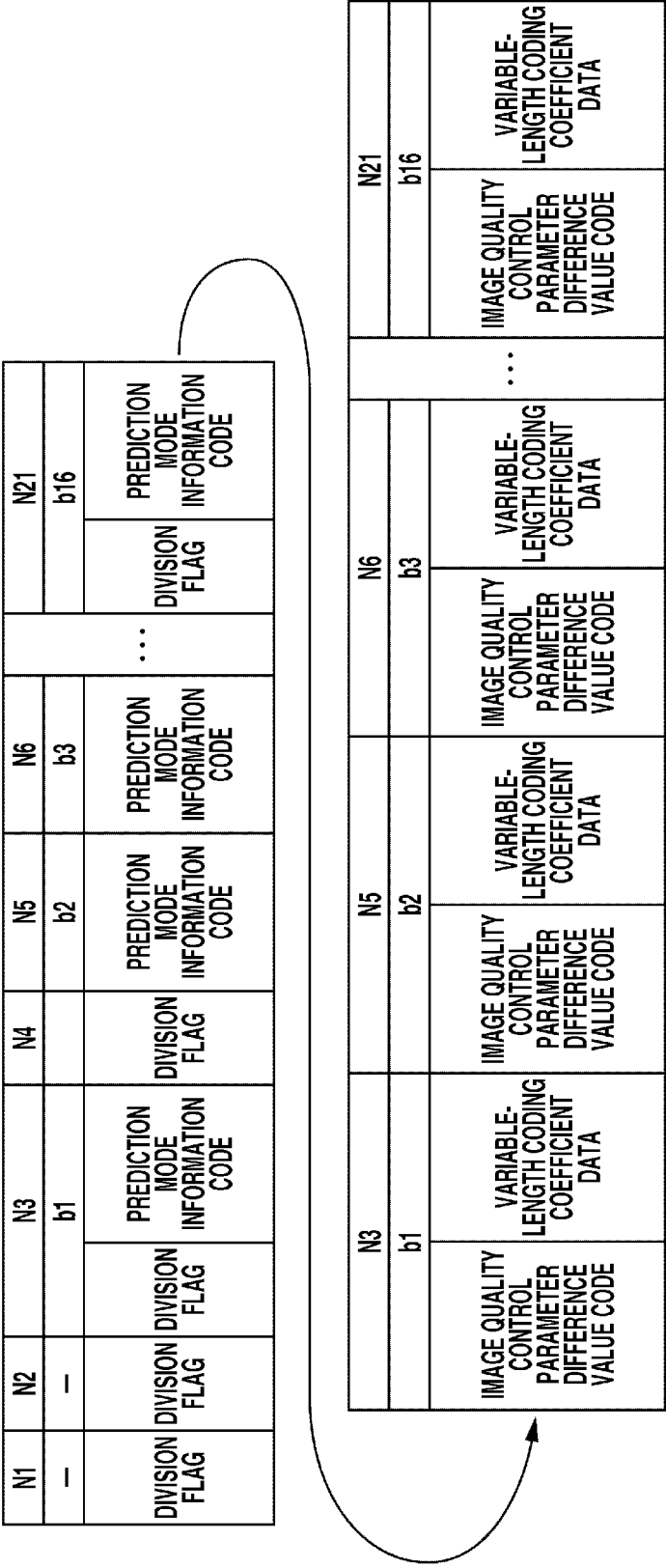
[Fig. 17]

EXAMPLE DATA STRUCTURE OF FRAME



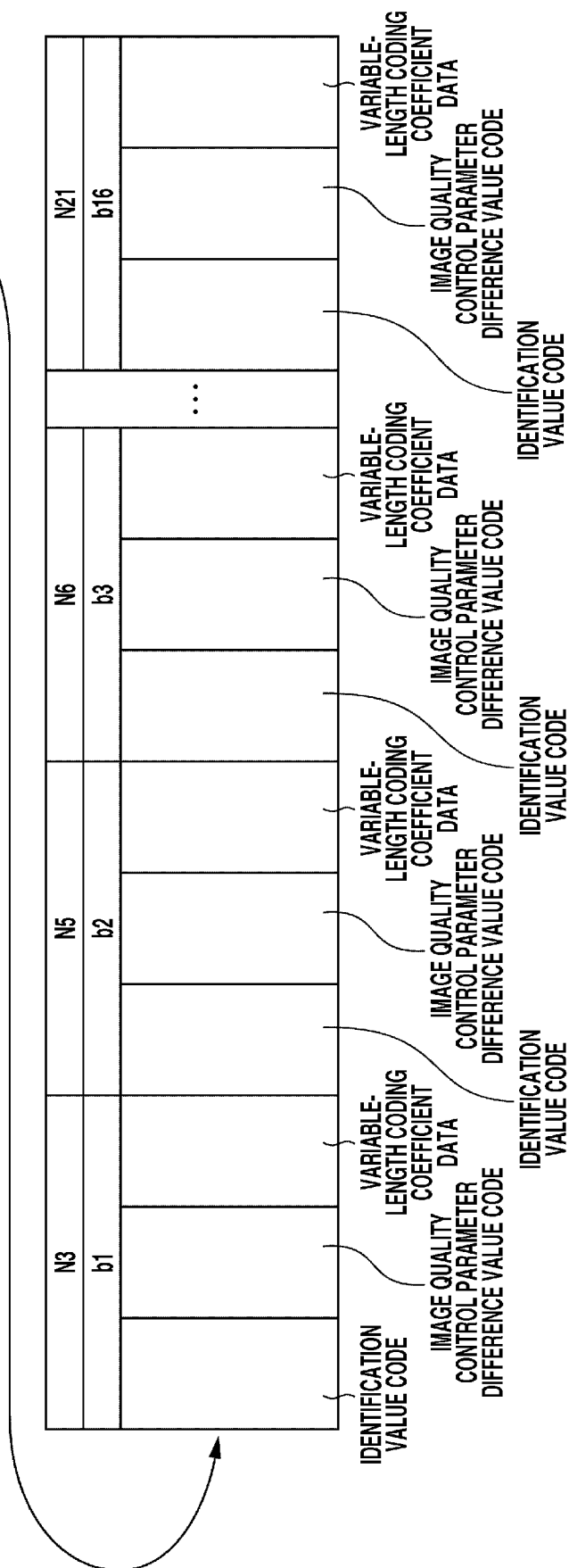
[Fig. 18A]

EXAMPLE DATA STRUCTURE OF LATTICE BLOCK

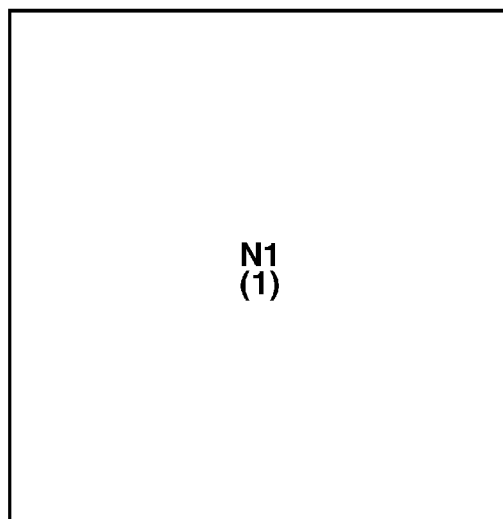


EXAMPLE DATA STRUCTURE OF LATTICE BLOCK

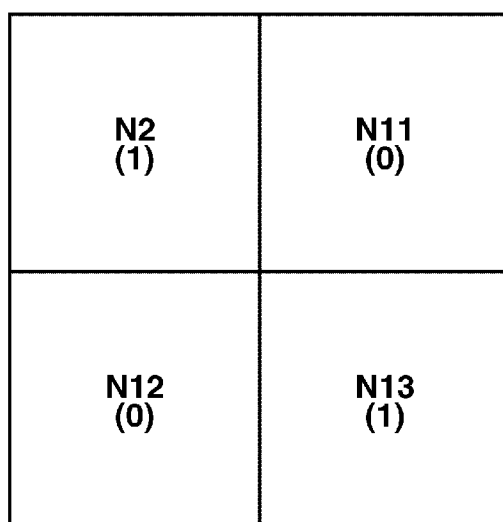
N1	N2	N3	N4	N5	N6	...	N21	
—	—	b1		b2	b3		b16	
DIVISION FLAG	DIVISION FLAG	DIVISION FLAG	DIVISION FLAG	PREDICTION MODE INFORMATION CODE	PREDICTION MODE INFORMATION CODE	PREDICTION MODE INFORMATION CODE	DIVISION FLAG	PREDICTION MODE INFORMATION CODE



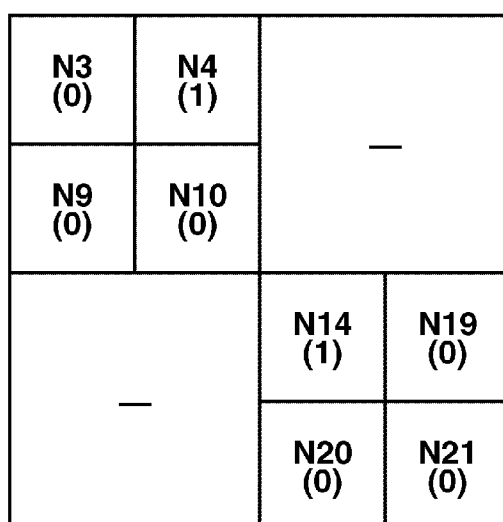
[Fig. 19A]



[Fig. 19B]



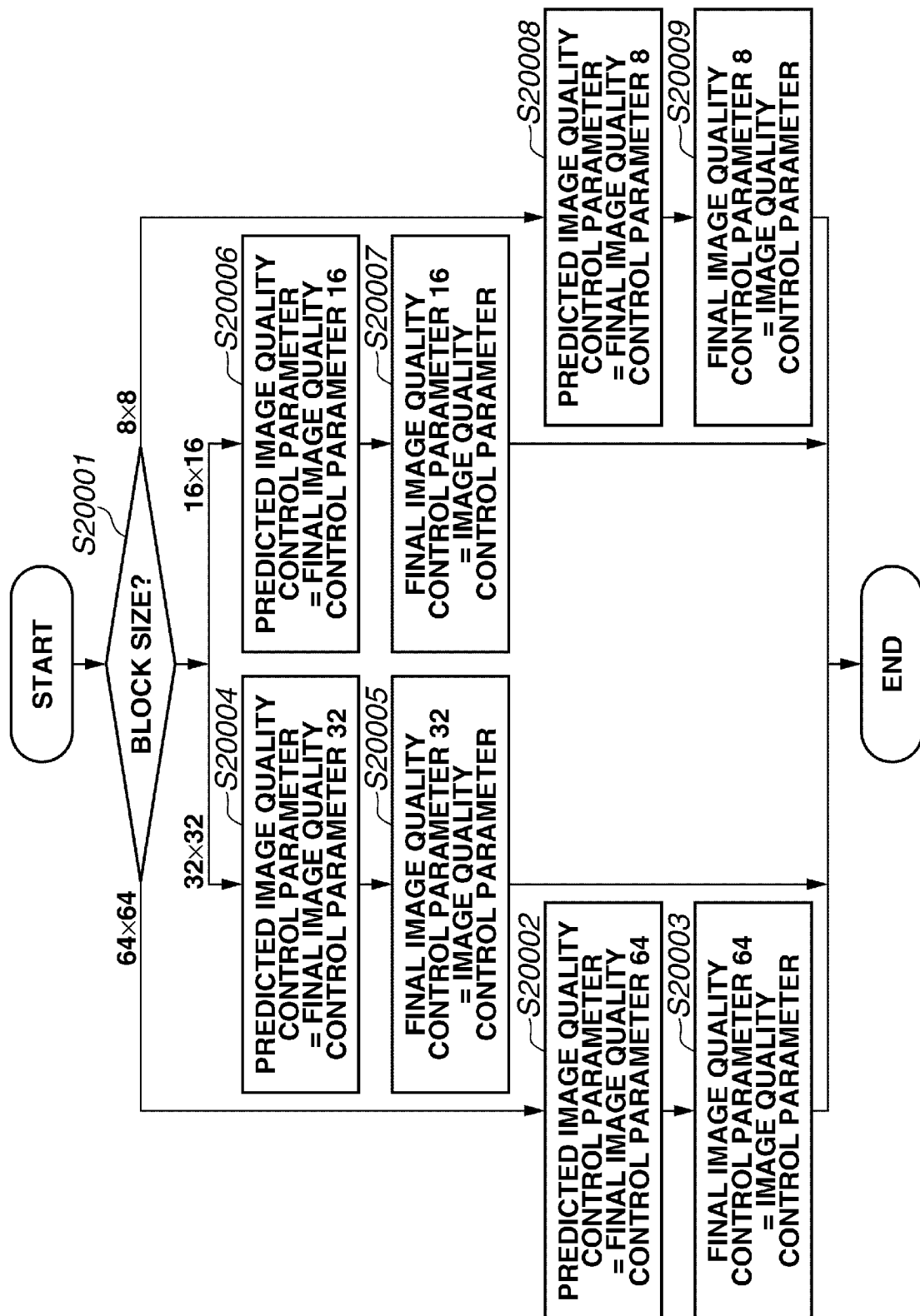
[Fig. 19C]



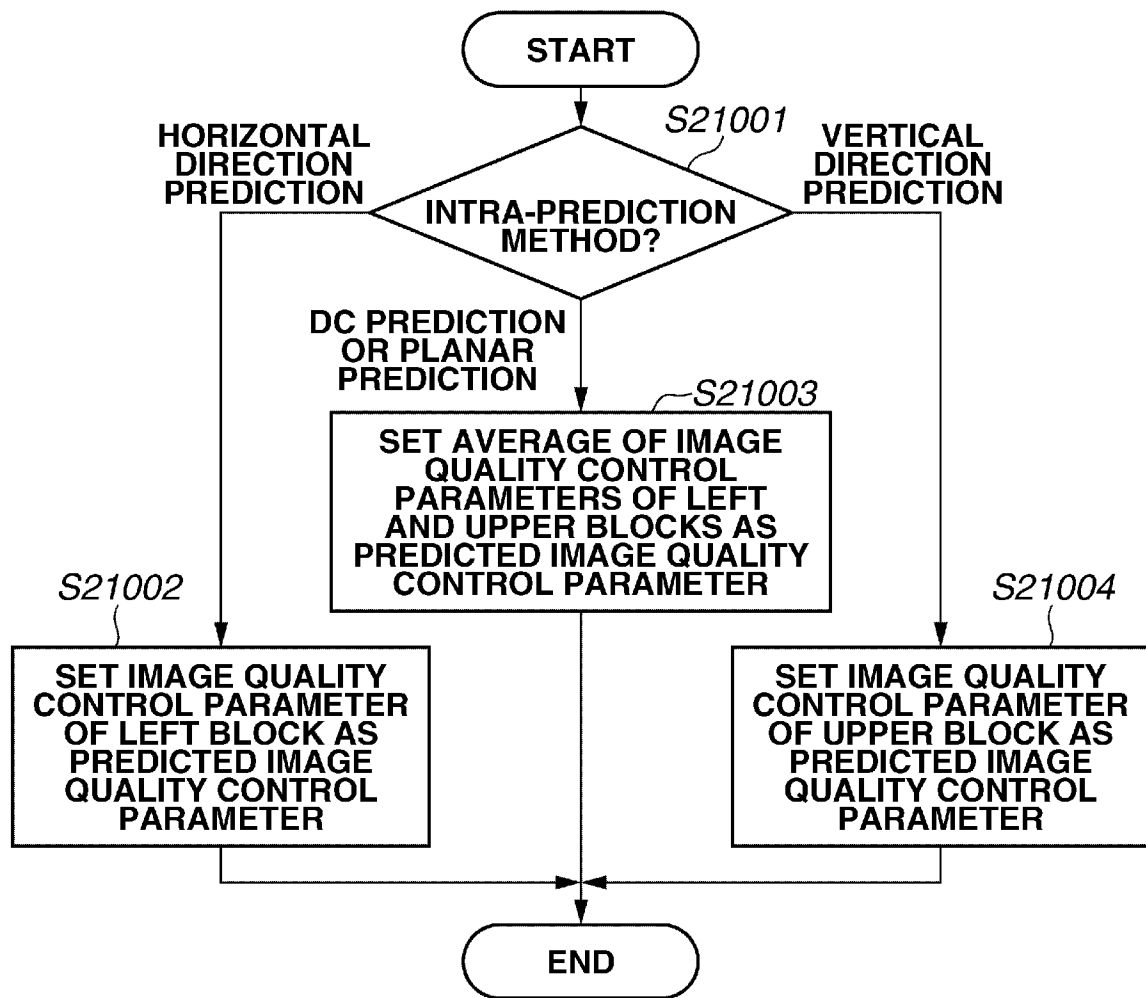
[Fig. 19D]

—	N5	N6	—		
	N7	N8			
—	—				
—			N15	N16	—
			N17	N18	
			—		—

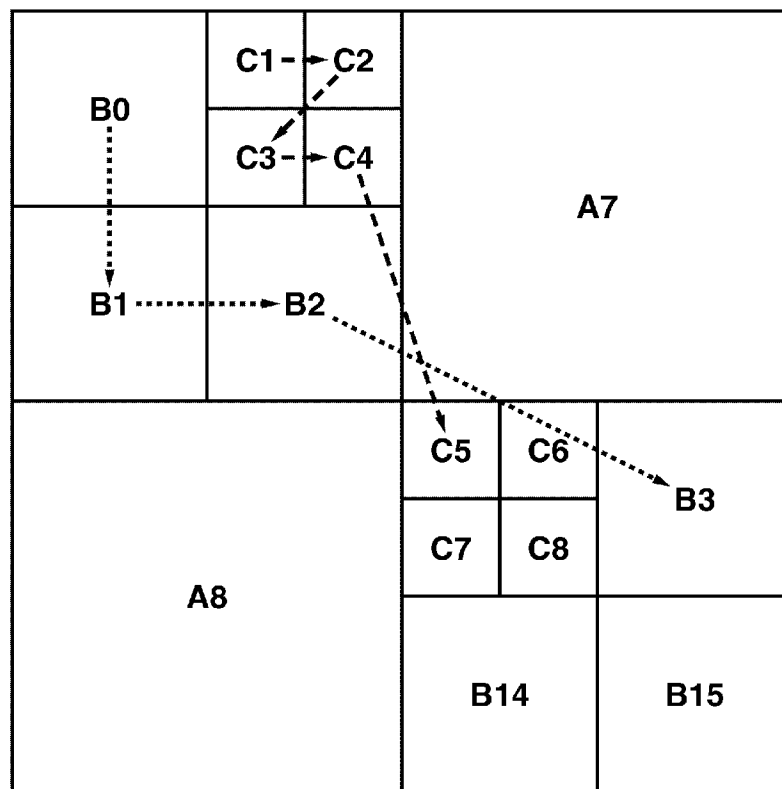
[Fig. 20]



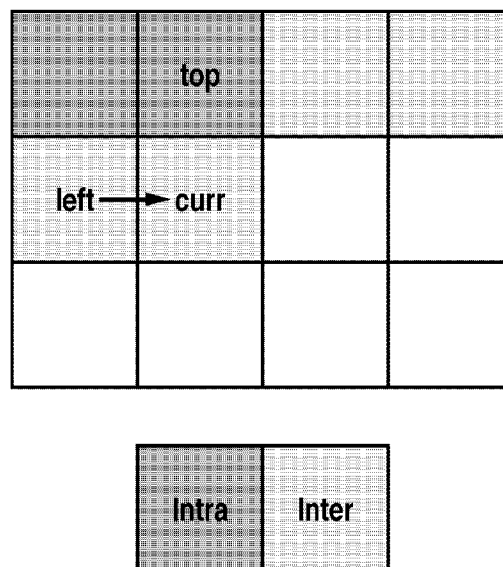
[Fig. 21]



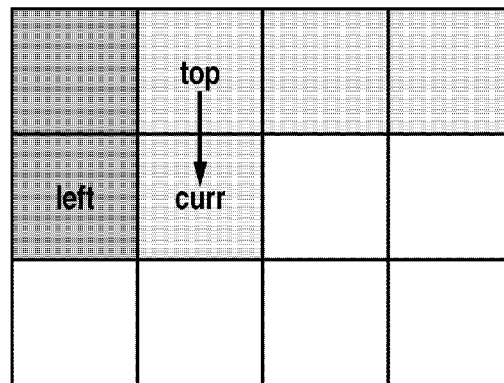
[Fig. 22]



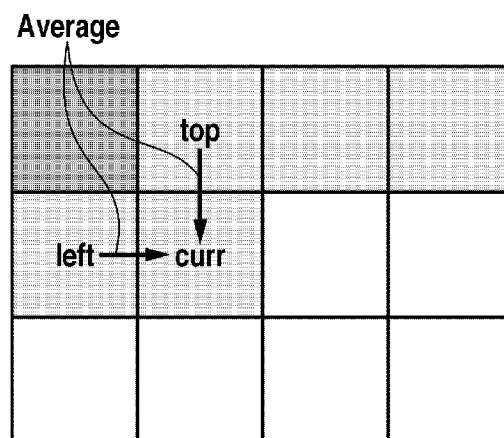
[Fig. 23A]



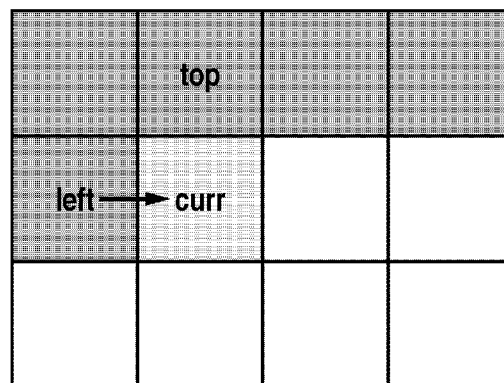
[Fig. 23B]



[Fig. 23C]



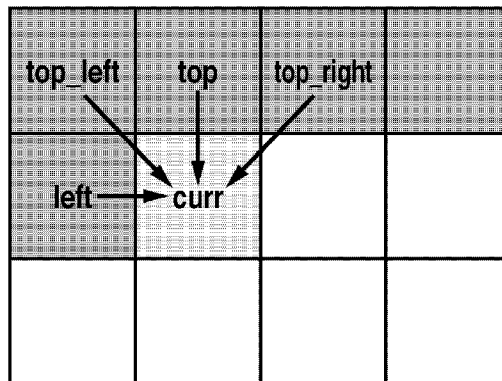
[Fig. 23D]



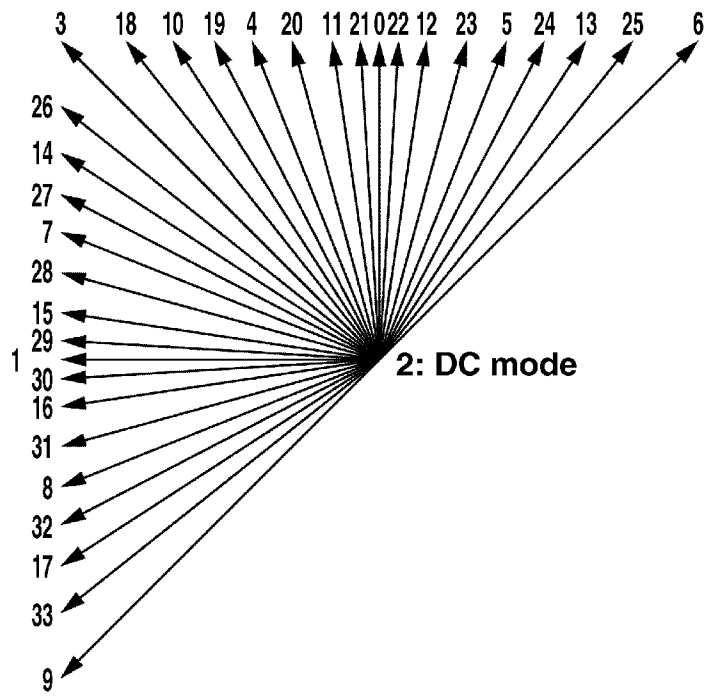
[Fig. 24A]

Equation	Intra Prediction Index
$\text{pred_qp}(\text{curr}) = \text{pred_qp}(\text{left})$	9,33,17,32,8,31,16,30,1,29,15,28,7
$\text{pred_qp}(\text{curr}) = (\text{qp}(\text{left}) + \text{qp}(\text{top_left}))/2$	27,14,26,3
$\text{pred_qp}(\text{curr}) = (\text{qp}(\text{top_left}) + \text{qp}(\text{top}))/2$	18,10,19
$\text{pred_qp}(\text{curr}) = \text{qp}(\text{top})$	4,20,11,21,0,22,12,23,5
$\text{pred_qp}(\text{curr}) = (\text{qp}(\text{left}) + \text{qp}(\text{top_left}))/2$	24,13,25,6
$\text{pred_qp}(\text{curr}) = (\text{qp}(\text{top}) + \text{qp}(\text{left}))/2$	2

[Fig. 24B]



[Fig. 25]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/001318

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H04N7/32 (2006.01) 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. H04N7/32

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 1971-2012
 Registered utility model specifications of Japan 1996-2012
 Published registered utility model applications of Japan 1994-2012

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.
P, X	Masaaki Kobayashi and Masato Shima, Sub-LCU level delta QP signaling, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 5th Meeting: Geneve, CH, Document: JCTVC-E198, ITU-T, 2011.03.10	1-4
A	Kazushi Sato, Proposal on Large Block Structure and Quantization, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 3rd Meeting: Guangzhou, CN, Document: JCTVC-C167, ITU-T, 2010.10	1-4

☒ 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

18.05.2012

Date of mailing of the international search report

29.05.2012

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

Takayuki HATANAKA

Telephone No. +81-3-3581-1101 Ext. 3541

5C

9468

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/001318

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Madhukar Budagavi and Minhua Zhou, Delta QP signaling at sub-LCU level, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting: Daegu, KR, Document: JCTVC-D038, ITU-T, 2011.01	1-4
A	L. dong et al., CU Adaptive Quantization Syntax Change for Better Decoder pipelining, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting: Daegu, KR, Document: JCTVC-D258, ITU-T, 2011.01	1-4
A	Kazushi Sato, On LBS and Quantization, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting: Daegu, KR, Document: JCTVC-D308r1, ITU-T, 2011.01	1-4
E, A	WO 2012/056924 A1 (SONY CORPORATION) 2012.05.03, pars. [0027] - [0028] , [0035] , [0109] - [0117] , [0179] - [0183] , Fig.6,15,21 (No Family)	1-4