

**IMAGE CODING DEVICE, IMAGE CODING METHOD, IMAGE CODING  
PROGRAM, IMAGE DECODING DEVICE, IMAGE DECODING METHOD AND  
IMAGE DECODING PROGRAM**

5 BACKGROUND OF THE INVENTION

[TECHNICAL FIELD]

**[0001]** The present invention relates to a technology of  
segmenting an image into blocks and encoding and decoding the  
10 image in units of blocks resulting from segmenting the image.

[BACKGROUND ART]

**[0002]** In image encoding and decoding, an image is  
segmented into blocks, each of which is a set of a  
predetermined number of pixels. The image is encoded and  
15 decoded in units of blocks. By performing proper block  
segmentation, the efficiency of intra-picture prediction  
(intra prediction), inter-picture prediction (inter  
prediction), orthogonal transform, entropy encoding, etc. is  
improved. As a result, the encoding efficiency is improved.

20 [CITATION LIST]

[PATENT LITERATURE]

**[0003]** [patent document 1] JP2015-526008

[DISCLOSURE OF THE INVENTION]

**[0004]** Unless an image is segmented into blocks of  
25 proper sizes and shapes, the encoding efficiency is reduced.  
If an image is not segmented into blocks of proper sizes and

shapes, the processing volume in the subsequent encoding and decoding will be increased.

**[0005]** The present invention addresses the  
aforementioned issue, and a purpose thereof is to provide a  
5 technology of improving the encoding efficiency by performing  
block segmentation suited to image encoding and decoding.

**[0006]** An image encoding device according to an  
embodiment is adapted to segment an image into blocks and  
encode the image in units of blocks resulting from segmenting  
10 the image and includes: a block segmentation unit (101) that  
recursively segments the image into rectangles of a  
predetermined size to generate a block subject to encoding;  
and an encoding unit (105) that encodes block segmentation  
information of the block subject to encoding. The block  
15 segmentation unit includes: a quartering unit that quarters a  
target block in recursive segmentation in a horizontal  
direction and a vertical direction to generate four blocks;  
and a halving unit that halves a target block in recursive  
segmentation in a horizontal direction or a vertical  
20 direction to generate two blocks. When previous recursive  
segmentation is halving, the halving unit prohibits a target  
block subject to current recursive segmentation from being  
segmented in the same direction as a direction in which the  
block was segmented in the previous recursive segmentation.

25 **[0007]** Another embodiment also relates to an image  
encoding device. The device is adapted to segment an image

into blocks and encode the image in units of blocks resulting from segmenting the image and includes: a block segmentation unit (101) that recursively segments the image into rectangles of a predetermined size to generate a block

5 subject to encoding; and an encoding unit (105) that encodes block segmentation information of the block subject to encoding. The block segmentation unit includes: a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to

10 generate four blocks; and a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks. When the last but one recursive segmentation is halving and previous recursive segmentation is quartering, the block segmentation

15 unit prohibits further segmentation of the target block.

**[0008]** Still another embodiment relates to an image encoding method. The method is adapted to segment an image into blocks and encode the image in units of blocks resulting from segmenting the image and includes: recursively

20 segmenting the image into rectangles of a predetermined size to generate a block subject to encoding; and encoding block segmentation information of the block subject to encoding. The segmenting includes: quartering a target block in recursive segmentation in a horizontal direction and a

25 vertical direction to generate four blocks; and halving a target block in recursive segmentation in a horizontal

direction or a vertical direction to generate two blocks.

When previous recursive segmentation is halving, the halving prohibits a target block subject to current recursive segmentation from being segmented in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

**[0009]** Still another embodiment relates to an image decoding device. The device is adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image and includes: a block segmentation unit (202) that recursively segments the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and a decoding unit (203, 204, 205) that decodes the block subject to decoding. The block segmentation unit includes: a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks. When previous recursive segmentation is halving, the decoding unit does not decode a flag indicating whether to segment a target block subject to current recursive segmentation in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

**[0010]** Still another embodiment also relates to an image decoding device. The device is adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image and includes: a block segmentation unit (202) that recursively segments the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and a decoding unit that (203, 204, 205) decodes the block subject to decoding. The block segmentation unit includes: a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks. When the last but one recursive segmentation is halving and previous recursive segmentation is quartering, the decoding unit does not decode a flag indicating whether to segment the target block further.

**[0011]** Still another embodiment relates to an image

decoding method. The method is adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image and includes: recursively segmenting the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and decoding the block subject to decoding. The segmenting includes: quartering a target block

in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and halving a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks.

5 When previous recursive segmentation is halving, the decoding does not decode a flag indicating whether to segment a target block subject to current recursive segmentation in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

10 **[0012]** Optional combinations of the aforementioned constituting elements, and implementations of the embodiments in the form of methods, apparatuses, systems, recording mediums, and computer programs may also be practiced as additional modes of the embodiments.

15 **[0013]** According to the embodiments, a block segmentation suited to image encoding and decoding can be performed resulting in improving the encoding efficiency and providing encoding and decoding an image with a less processing volume.

20 [BRIEF DESCRIPTION OF THE DRAWINGS]

**[0014]** Fig. 1 shows a configuration of an image encoding device according to the first embodiment;

Fig. 2 shows a configuration of an image decoding device according to the first embodiment;

25 Fig. 3 is a flowchart showing segmentation into tree blocks and segmentation within a tree block;

Fig. 4 shows how an input image is segmented into tree blocks;

Fig. 5 shows z-scan;

Fig. 6 shows a tree block quartered in the horizontal  
5 and vertical directions;

Fig. 7 shows a tree block halved in the horizontal  
direction;

Fig. 8 shows a tree block halved in the vertical  
direction;

10 Fig. 9 is a flowchart showing a process in each block  
resulting from quartering a tree block in the horizontal  
direction and the vertical direction;

Fig. 10 is a flowchart showing a process in each block  
resulting from halving a tree block in the horizontal  
15 direction;

Fig. 11 shows how a block resulting from halving a tree  
block in the horizontal direction is re-segmented;

Fig. 12 is a flowchart showing a process in each block  
resulting from halving a tree block in the vertical  
20 direction;

Fig. 13 shows how a block resulting from halving a tree  
block in the vertical direction is re-segmented;

Fig. 14 shows an example of syntax related to block  
segmentation according to the first embodiment;

25 Figs. 15A-15D show intra prediction;

Fig. 16 shows inter prediction;

Fig. 17 shows an example of syntax related to block segmentation according to the second embodiment;

Fig. 18 shows another example of syntax related to block segmentation according to the second embodiment;

5        Fig. 19 shows an example of syntax related to block segmentation according to the third embodiment;

Fig. 20 shows how a block resulting from halving a parent block in the horizontal direction or the vertical direction is further segmented inside in the same direction;

10       Fig. 21 shows an example of syntax related to block segmentation according to the fourth embodiment; and

Fig. 22 shows quartering of a block resulting from halving a tree block.

[BEST MODE FOR CARRYING OUT THE INVENTION]

15    **[0015]**        Embodiments of the present invention provide an image encoding technology of segmenting an image into rectangular blocks and encoding and decoding a segment block.

**[0016]**

(First embodiment)

20       A description will be given of an image encoding device 100 and an image decoding device 200 according to the first embodiment. In the first embodiment, a block is restricted from being segmented successively in the same direction recursive block segmentation.

25    **[0017]**        Fig. 1 shows a configuration of the image encoding device 100 according to the first embodiment; It

should be noted that Fig. 1 only shows a flow of data related to an image signal. Additional information (e.g., a motion vector, a prediction mode, etc.) other than an image signal is supplied by relevant constituting elements to a bitstream generation unit 105 to generate associated encoded data, but the figure does not show a flow of data related to the additional information.

**[0018]** The block segmentation unit 101 segments an image into blocks subject to encoding, each of which is a unit of processing in encoding, and supplies an image signal in the block subject to encoding to a residual signal generation unit 103. Further, the block segmentation unit 101 supplies an image signal of a block subject to encoding to a predicted image generation unit 102 to evaluate a matching score of a predicted image.

**[0019]** The block segmentation unit 101 recursively segments an image into rectangles of a predetermined size to generate blocks subject to encoding. The block segmentation unit 101 includes a quartering unit that quarters a target block in recursive segmentation in the horizontal direction and the vertical direction to generate four blocks and a halving unit that halves a target block in recursive segmentation in the horizontal direction or the vertical direction to generate two blocks. The detailed operation of the block segmentation unit 101 will be described later.

**[0020]** The predicted image generation unit 102 performs

intra-picture prediction (intra prediction) or inter-picture prediction (inter prediction) from a decoded image signal supplied from a decoded image memory 108 by referring to a prediction mode, thereby generating a predicted image signal.

5 The image signal in the block subject to encoding supplied from the block segmentation unit 101 is used for evaluation of intra prediction and inter prediction. In intra prediction, a predicted image signal is generated by using the image signal of the block subject to encoding supplied  
10 from the block segmentation unit 101 and an image signal, supplied from the decoded image memory 108, of an encoded block neighboring the block subject to encoding and located in the same picture as the block subject to encoding. In inter prediction, an encoded picture, located before or after  
15 a picture including the block subject to encoding (encoding picture) in the time series and stored in the decoded image memory 108, is defined as a reference picture. An image signal of the block subject to encoding supplied from the block segmentation unit 101 is subject to evaluation for the  
20 degree of block-to-block coincidence (for example, subject to block matching) between the encoding picture and the reference picture. A motion vector indicating an amount of movement is determined, and a predicted image signal is generated by performing motion compensation based on the  
25 amount of movement from the reference picture. The predicted image generation unit 102 supplies the predicted image signal

thus generated to the residual signal generation unit 103.

**[0021]** The residual signal generation unit 103 subtracts the predicted signal generated by the predicted image generation unit 102 from the image signal subject to encoding  
5 to generate a residual signal and supplies the residual signal to an orthogonal transform/quantization unit 104.

**[0022]** The orthogonal transform/quantization unit 104 subjects the residual signal supplied from the residual signal generation unit 103 to orthogonal transform and  
10 quantization and supplies the orthogonally transformed and quantized residual signal to the bitstream generation unit 105 and an inverse-quantization/inverse-orthogonal transform unit 106.

**[0023]** The bitstream generation unit 105 generates a  
15 bitstream corresponding to the orthogonally transformed and quantized residual signal supplied from the orthogonal transform/quantization unit 104. The bitstream generation unit 105 also generates a bitstream corresponding to additional information such as the motion vector, prediction  
20 mode, and block segmentation information.

**[0024]** The inverse-quantization/inverse-orthogonal transform unit 106 subjects the orthogonally transformed and quantized residual signal supplied from the orthogonal transform/quantization unit 104 to inverse-  
25 quantization/inverse-orthogonal transform and supplies the residual signal subjected to inverse-quantization and

inverse-orthogonal transform to a decoded image signal  
superimposition unit 107.

**[0025]** The decoded image signal superimposition unit 107  
superimposes the predicted image signal generated by the  
5 predicted image generation unit 102 and the residual signal  
subjected to inverse-quantization and inverse-orthogonal  
transform by the inverse-quantization/inverse-orthogonal  
transform unit 106 one upon the other so as to generate a  
decoded image. The decoded image signal superimposition unit  
10 107 stores the decoded image in the decoded image memory 108.  
The decoded image may be subject to filtering for reducing  
block distortion, etc. resulting from encoding before being  
stored in the decoded image memory 108.

**[0026]** Fig. 2 shows a configuration of an image decoding  
15 device 200 according to the first embodiment. It should be  
noted that Fig. 2 only shows a flow of data related to an  
image signal. Additional information (e.g., a motion vector,  
a prediction mode, etc.) other than an image signal is  
supplied by a bitstream decoding unit 201 to relevant  
20 constituting elements and used in an associated process, but  
the figure does not show a flow of data related to the  
additional information.

**[0027]** The bitstream decoding unit 201 decodes the  
supplied bitstream and supplies the orthogonally transformed  
25 and quantized residual signal to a block segmentation unit  
202.

**[0028]**        The block segmentation unit 202 determines the shape of a block subject to decoding based on the decoded block segmentation information and supplies the orthogonally transformed and quantized residual signal of the block subject to decoding thus determined to an inverse-quantization/inverse-orthogonal transform unit 203.

**[0029]**        The block segmentation unit 202 recursively segments the image into rectangles of a predetermined size based on the decoded block segmentation information to generate a block subject to decoding. The block segmentation unit 202 includes a quartering unit that quarters a target block in recursive segmentation in the horizontal direction and the vertical direction to generate four blocks and a halving unit that halves a target block in recursive segmentation in the horizontal direction or the vertical direction to generate two blocks. The detailed operation of the block segmentation unit 202 will be described later.

**[0030]**        The inverse-quantization/inverse-orthogonal transform unit 203 subjects the orthogonally transformed and quantized residual signal supplied to inverse-quantization and inverse-orthogonal transform so as to obtain a residual signal subjected to inverse-quantization and inverse-orthogonal transform.

**[0031]**        The predicted image generation unit 204 generates a predicted image signal from the decoded image signal supplied from a decoded image memory 206 and supplies the

predicted image to a decoded image signal superimposition unit 205.

**[0032]** The decoded image signal superimposition unit 205

superimposes predicted image signal generated by the

5 predicted image generation unit 204 and the residual signal

subjected to inverse-quantization and inverse-orthogonal

transform by the inverse-quantization/inverse-orthogonal

transform unit 203 one upon the other so as to generate and

output a decoded image signal. The decoded image signal

10 superimposition unit 205 stores the decoded image signal in

the decoded image memory 206. The decoded image may be

subject to filtering for reducing block distortion, etc.

resulting from encoding before being stored in the decoded

image memory 206.

15 **[0033]** A detailed description will be given of the

operation of the block segmentation unit 101 of the image

encoding device 100. Fig. 3 is a flowchart showing

segmentation into tree blocks and segmentation within a tree

block.

20 **[0034]** First, an input image is segmented into tree

blocks of a predetermined size (S1000). For example, a tree

block includes 128 pixels × 128 pixels. However, a tree

block need not necessarily include 128 pixels × 128 pixels.

Any size and shape may be used so long as the tree block is

25 rectangular. The fixed values defining the size and shape of

a tree block may be established in the encoding device and

the decoding device. The encoding device may determine the size and shape and record the size and shape in the bitstream, and the decoding device may use the recorded block size. Fig. 4 shows how an input image is segmented into tree blocks.

5 Tree blocks are encoded and decoded in the order of raster scan, i.e., from left to right and from top to bottom.

**[0035]** A tree block is further segmented inside into rectangular blocks. A tree block is encoded and decoded inside in the order of z-scan. Fig. 5 shows the order of z-scan. In z-scan, the top left block, the top right block, the bottom left block, and the bottom right block are encoded and decoded in the stated order. A tree block can be quartered or halved inside. In quartering, a tree block is segmented in the horizontal direction and the vertical  
10 direction. In halving, a tree block is segmented in the horizontal direction or the vertical direction. Fig. 6 shows a tree block quartered in the horizontal and vertical directions. Fig. 7 shows a tree block halved in the horizontal direction. Fig. 8 shows a tree block halved in  
15 the vertical direction.

**[0036]** Reference is made back to Fig. 3. A determination is made as to whether to quarter a tree block inside in the horizontal direction and the vertical direction (S1001).

25 **[0037]** When it is determined to quarter the tree block inside (S1001: Yes), the tree block is quartered inside

(S1002), and the blocks resulting from quartering the tree block in the horizontal and vertical directions are processed respectively (S1003). Re-segmentation of the blocks resulting from quartering the tree block will be described later (Fig. 9).

**[0038]** When it is determined not to quarter the tree block inside (S1001: No), a determination is made as to whether to halve the tree block inside (S1004).

**[0039]** When it is determined to halve the tree block inside (S1004: Yes), a determination is made as to whether to halve the tree block in the horizontal direction (S1005).

**[0040]** When it is determined to halve the block in the horizontal direction (S1005: Yes), the tree block is halved inside in the horizontal direction (S1006), and the blocks resulting from halving the tree block in the horizontal direction are processed respectively (S1007). Re-segmentation of the blocks resulting from halving the tree block in the horizontal direction will be described later (Fig. 10).

**[0041]** When it is determined to halve the tree block in the vertical direction instead of the horizontal direction (S1005: No), the tree block is halved inside in the vertical direction (S1008), and the blocks resulting from halving the tree block in the vertical direction are processed respectively (S1009). Re-segmentation of the blocks resulting from halving the tree block in the vertical

direction will be described later (Fig. 11).

**[0042]** When it is determined not to halve the tree block inside (S1004: No), the tree block is not segmented inside and the block segmentation process is terminated (S1010).

5 **[0043]** A description will now be given of a process of each block resulting from quartering the tree block in the horizontal and vertical directions with reference to the flowchart of Fig. 9.

**[0044]** A determination is made as to whether to quarter  
10 a block inside again in the horizontal and vertical directions (S1101).

**[0045]** When it is determined to quarter the block inside again (S1101: Yes), the block is quartered inside again (S1102), and the blocks resulting from quartering the block  
15 in the horizontal and vertical directions are processed respectively (S1103).

**[0046]** When it is determined not to quarter the block inside again (S1101: No), a determination is made as to whether to halve the block inside (S1104).

20 **[0047]** When it is determined to halve the block inside (S1104: Yes), a determination is made as to whether to halve the block in the horizontal direction (S1105).

**[0048]** When it is determined to halve the block in the horizontal direction (S1105: Yes), the block is halved inside  
25 in the horizontal direction (S1106), and the blocks resulting from halving the block in the horizontal direction are

processed respectively (S1107).

**[0049]** When it is determined to halve the lock in the vertical direction instead of the horizontal direction (S1105: No), the block is halved inside in the vertical direction (S1108), and the blocks resulting from halving the block in the vertical direction are processed respectively (S1109).

**[0050]** When it is determined not to halve the block inside (S1104: No), the block is not segmented inside and the block segmentation process is terminated (S1110).

**[0051]** The process shown in the flowchart of Fig. 9 is performed for each block resulting from the quartering. The blocks resulting from quartering a block is also encoded and decoded inside in the order of z-scan.

**[0052]** A description will now be given of the process of each block resulting from halving a tree block in the horizontal direction with reference to the flowchart of Fig. 10.

**[0053]** When a tree block is halved in the horizontal direction, a determination is made as to whether to quarter each block resulting from halving the tree block in the horizontal and vertical directions inside (S1201).

**[0054]** When it is determined to quarter the block inside (S1201: Yes), the block is quartered inside (S1202), and the blocks resulting from quartering the block in the horizontal and vertical directions are processed respectively (S1203).

**[0055]** When it is determined not to quarter the block inside (S1201: No), a determination is made as to whether to halve the block inside again (S1204).

**[0056]** When it is determined to halve the block again (S1204: Yes), the block is halved inside in the vertical direction (S1205), and the blocks resulting from halving the block in the vertical direction are processed respectively (S1206).

**[0057]** When it is determined not to halve the block again (S1204: No), the block is not segmented inside again and the block segmentation process is terminated (S1207).

**[0058]** Fig. 11 shows how a block resulting from halving a tree block in the horizontal direction is re-segmented. In this case, when a tree block (parent block) is halved in the horizontal direction, the re-halving of the segment block is permitted only in the vertical direction, and the block is halved in the vertical direction automatically. Quartering of a child block resulting from halving a tree block (parent block) may be completely prohibited. This prohibits a block from being segmented in the same direction as the parent block and so prevents block segmentation that results in a rectangle further elongated in the lateral direction, facilitating the encoding and decoding process.

**[0059]** The process shown in the flowchart of Fig. 10 is performed for each block resulting from halving a parent block in the horizontal direction. The blocks resulting from

halving a parent block is also encoded and decoded inside from top to bottom.

**[0060]** A description will now be given of the process of each block resulting from halving a tree block in the

5 vertical direction with reference to the flowchart of Fig. 12.

**[0061]** When a tree block is halved in the vertical direction, a determination is made as to whether to quarter each block resulting from halving the tree block is quartered in the horizontal and vertical directions inside (S1301).

10 **[0062]** When it is determined to quarter the block inside (S1301: Yes), the block is quartered inside (S1302), and the blocks resulting from quartering the block in the horizontal and vertical directions are processed respectively (S1303).

**[0063]** When it is determined not to quarter the block  
15 inside (S1301: No), a determination is made as to whether to halve the block inside again (S1304).

**[0064]** When it is determined to halve the block again (S1304: Yes), the block is halved inside in the horizontal direction (S1305), and the blocks resulting from halving the  
20 block in the horizontal direction are processed respectively (S1306).

**[0065]** When it is determined not to halve the block again (S1304: No), the block is not segmented inside again and the block segmentation process is terminated (S1307).

25 **[0066]** Fig. 13 shows how a block resulting from halving a tree block in the vertical direction is re-segmented.

In this case, when a tree block (parent block) is halved in the vertical direction, the re-halving of the segment block is permitted only in the horizontal direction, and the block is halved in the horizontal direction automatically.

- 5   Quartering of a child block resulting from halving a tree block (parent block) may be completely prohibited. This prohibits a block from being segmented in the same direction as the parent block and so prevents block segmentation that results in a rectangle further elongated in the vertical  
10   direction, facilitating the encoding and decoding process.

**[0067]**       The process shown in the flowchart of Fig. 12 is performed for each block resulting from halving a parent block in the vertical direction. The blocks resulting from halving a parent block are also encoded and decoded inside  
15   from left to right.

**[0068]**       While re-segmentation of a block resulting from segmenting a tree block has been described above, a parent block need not be a tree block. For example. the above process may be applied to the segmentation of a block  
20   resulting from quartering a tree block (128X128) and further quartering or halving the block (64X64) resulting from quartering the tree block.

**[0069]**       A description will now be given of the operation of the block segmentation unit 202 of the image decoding  
25   device 200. The block segmentation unit 202 segments a block through the same processing steps as performed in the block

segmentation unit 101 of the image encoding device 100. The block segmentation unit 101 of the image encoding device 100 selects a pattern of block segmentation and outputs block segmentation information indicating the selection. The block segmentation unit 202 of the image decoding device differs in that it segments a block by using the block segmentation information retrieved from the bitstream by decoding the bitstream and employs a syntax structure in which the block segmentation information is retrieved from the bitstream by decoding the bitstream such that the information that lacks an option is not transferred in the bitstream when re-segmentation in the same direction is prohibited.

**[0070]** Fig. 14 shows an example of syntax (syntactic

rule of a bitstream) related to block segmentation according

to the first embodiment. For segmentation of a tree block

inside, a flag (4\_division\_flag) indicating whether to

quarter the tree block is first transmitted and received. In

the case the tree block is quartered (4\_division\_flag is set

to 1), the tree block is quartered and the process is

terminated. The block resulting from quartering the tree

block is then re-segmented inside again according to the

syntax shown in Fig. 14. In the case the tree block is not

quartered (4\_division\_flag is set to 0), a flag

(2\_division\_flag) indicating whether to halve the tree block

is transmitted and received. In the case the tree block is

halved (2\_division\_flag is set to 1), a flag

(2\_division\_direction) indicating the direction of halving is transmitted and received. 2\_division\_direction set to 1 means segmentation in the vertical direction, and 2\_division\_direction set to 0 means segmentation in the horizontal direction. The block resulting from halving the tree block is then re-segmented inside again according to the syntax shown in Fig. 14. In the case the tree block is not halved (2\_division\_flag is set to 0), the tree block is not segmented, and the process is terminated.

10 **[0071]** A description will now be given of the process of re-segmenting the block resulting from quartering or halving a block will be described. The process of re-segmenting the block inside also uses the syntax shown in Fig. 14 but differs from the process of segmenting a tree block in that

15 the direction of halving is restricted. Specifically, when a tree block is halved and the block resulting from halving the tree block is re-segmented, segmentation in the same direction as the direction of halving the tree block is prohibited. This prevents a block resulting from the

20 segmentation from being a further elongated rectangle, thereby preventing an increase in the memory bandwidth necessary for intra prediction and inter prediction. Details of prevention of an increase in the memory bandwidth will be discussed later.

25 **[0072]** Of course, the number of times of halving in the same direction may be counted, and segmentation in the same

direction may be restricted when the number of times exceeds a predetermined count. For example, halving in the same direction may be permitted twice, but halving in the same direction for the third time may be prohibited.

5   **[0073]**       Referring to the syntax of Fig. 14, quartering is selected preferentially, and the information indicating whether to quarter a block is transmitted and received before the information indicating whether to halve the block. In the case halving is selected preferentially, on the other

10   hand, the syntax may be such that the information indicating whether to halve a block is transmitted and received before the information indicating whether to quarter the block. This is because the volume of the code transmitted in a bitstream will be smaller by transmitting and receiving an

15   event that is more likely to occur. In other words, an estimation is made in advance to see which of quartering and halving is more likely to occur and the syntax is configured to first transmit and receive segmentation information indicating the segmentation that is more likely to occur.

20   For example, by indicating whether to prioritize quartering or prioritize halving when transmitting and receiving the header information of an image, the encoding device can adaptively determine a prioritized segmentation type (the number of segment blocks) that will result in a higher coding

25   efficiency, and the decoding device can segment a tree block according to the syntax based on the selected prioritized

segmentation type.

**[0074]** In the image encoding device 100 and the image decoding device 200, intra prediction or inter prediction is performed by using segmented blocks. Both intra prediction and inter prediction accompany copying of pixels from a memory.

**[0075]** Figs. 15A-15D show an example of intra prediction. Figs. 15A and 15B show prediction directions and mode numbers for intra prediction. In intra prediction, a predicted image of a block subject to encoding or decoding is generated by copying pixels from the encoded or decoded pixels in proximity to the block subject to encoding or decoding, as shown in Fig. 15C and 15D. In intra prediction, generation of a predicted image and generation of encoded or decoded pixels are repeated in units of blocks. Therefore, the process proceeds sequentially in units of blocks so that the smaller a block is segmented inside, the larger the load from the whole process. Also, the more elongated the shape of a rectangular block, the heavier the process of copying pixels from the memory. Encoding and decoding requires orthogonal transform of a residual signal. Therefore, the larger the types of sizes of rectangles, the more numerous the necessary types of orthogonal transform and the larger the resultant circuit scale. Therefore, the memory bandwidth necessary for intra prediction is prevented from increasing when a block is halved inside, by restricting the block from being halved in

the same direction as the direction in which the parent block is segmented.

**[0076]** Fig. 16 shows an example of inter prediction. In

inter prediction, a predicted image of a block subject to

5 encoding or decoding is generated by copying, in units of

blocks, pixels from the pixels included in the encoded or

decoded image. Inter prediction often requires a device

configuration in which pixels are copied from a reference

image in units of blocks such that acquisition in units of

10 memory management including necessary pixels is required.

Therefore, the smaller a block is segmented and the more

elongated the shape of a rectangular block, the larger the

load from the whole process. Further, motion compensation of

decimal accuracy in a reference image using an interpolation

15 filter requires copying several pixels in addition to the

pixels included in the block. Therefore, the smaller the

size of a block, the larger the relative proportion of the

several pixels added and the larger the load from the whole

process. Accordingly, the memory bandwidth necessary for

20 intra prediction is prevented from increasing when a block is

halved inside, by restricting the block from being halved in

the same direction as the direction in which the parent block

is segmented.

**[0077]**

25 (Second embodiment)

A description will be given of an image encoding device

and an image decoding device according to the second embodiment of the present invention. The second embodiment differs from the first embodiment in that, when a block is of a predetermined size or smaller, further segmentation of the block inside is restricted and is identical to the first embodiment in the other respects. This prevents the load from the whole process from continuing to increase as a block is segmented inside into smaller blocks.

**[0078]** Figs. 17 and 18 show a syntax related to block

segmentation according to the second embodiment. The difference from the syntax of Fig. 14 according to the first embodiment is that block segmentation is enabled only when the size of the initial block is larger than a predetermined size. In the case of Fig. 17, a block including more than 64 pixels inside can be quartered or halved.

**[0079]** In the case a difference between the number of pixels in a block resulting from quartering and that of a block resulting from halving is considered, quartering is permitted when the number of pixels in the block is larger

than 64, and halving is permitted when the number of pixels in the block is larger than 32. In this way, restriction on the number of pixels in the block resulting from the segmentation is precisely controlled.

**[0080]**

(Third embodiment)

A description will be given of an image encoding device

and an image decoding device according to the third embodiment. The third embodiment differs from the first embodiment in that further vertical segmentation of a block resulting from vertical segmentation is restricted and is  
5 identical to the first embodiment in the other respects.

**[0081]** Normally, pixel information of an image is stored in a one-dimensional memory in the order of raster scan. In a one-dimensional memory, pixels in the horizontal direction are stored relatively close to each other, and pixels in the  
10 vertical direction are stored relatively remote from each other. It is therefore easy to access pixels in the horizontal direction, but it is not easy to access pixels in the vertical direction. In the case of a block including 16 horizontal pixels  $\times$  8 vertical pixels and a block including 8  
15 horizontal pixels  $\times$  16 vertical pixels, for example, the numbers of pixels are identical, but the block of 8 horizontal pixels  $\times$  16 vertical pixels consumes a more extensive range of the memory to store the pixels than the block of 16 horizontal pixels  $\times$  8 vertical pixels. Therefore,  
20 a larger memory bandwidth will be required for transfer of pixels when motion compensation is used.

**[0082]** Fig. 19 shows a syntax related to block segmentation according to the third embodiment. The difference from the syntax of Fig. 14 according to the first  
25 embodiment is that further vertical segmentation inside is

prohibited only when a parent block is halved vertically. Specifically, when the parent block is halved vertically and the child block should be halved inside, selection between horizontal and vertical is not available, and horizontal segmentation is automatically selected, as shown in Fig. 20.

**[0083]**

(Fourth embodiment)

A description will be given of an image encoding device and an image decoding device according to the fourth embodiment of the present invention. The fourth embodiment differs from the first embodiment in that, when a block is halved and then the resultant block inside is quartered, further segmentation of the resultant block inside is prohibited and is identical to the first embodiment in the other respects.

**[0084]** Fig. 21 shows a syntax related to block segmentation according to the fourth embodiment. As shown in Fig. 22, when a parent block is halved and then the resultant block inside is quartered, `2_division_after_4_division_flag` is set to 1, prohibiting all subsequent segmentation.

**[0085]** That a block is halved and then quartered means that non-selection of quartering when halving is selected has been established. Therefore, the likelihood that further block segmentation is necessary after the block is halved and then quartered is low. In the case such further segmentation is necessary, quartering may be selected from the outset. In

the case a block is halved and then quartered, the resultant block will already have a rectangular shape. Therefore, an attempt to prohibit the direction of halving after the quartering will complicate the process. By uniformly

5 prohibiting subsequent segmentation of a block resulting from halving and subsequent quartering, a determination as to whether block segmentation is enabled is prevented from becoming complicated. Uniform prohibition of further block segmentation in a block resulting from halving and subsequent  
10 quartering makes it unnecessary to transmit and receive the selection not to segment the block and reduces the volume of the code transmitted.

**[0086]** It will of course be possible to combine a plurality of the schemes for restriction of block  
15 segmentation according to first through fourth embodiments.

**[0087]** The bitstream of images output from the image encoding device according to any of the embodiments described above has a predefined format so that it can be decoded in accordance with the encoding method used in the embodiments.  
20 The image decoding device compatible with the image encoding device is capable of decoding the bitstream of the predefined data format.

**[0088]** If a wired or wireless network is used to exchange bitstreams between the image encoding device and the  
25 image decoding device, the bitstream may be converted into a data format suited to the mode of transmission over the

communication channel and be transmitted accordingly. In this case, there should be provided a transmitting device for converting the bitstreams output from the image encoding device into encoded data of a data format suited to the mode of transmission over the communication channel and for transmitting the bitstreams over the network, and a receiving device for receiving the bitstreams from the network to recover the bitstreams and supplying the recovered bitstreams to the image decoding device.

10   **[0089]**       The transmitting device includes a memory for buffering bitstreams output from the image encoding device, a packet processing unit for packetizing the bitstreams, and a transmitting unit for transmitting the packetized bitstreams over the network. The receiving device includes a receiving unit for receiving the packetized encoded data over the network, a memory for buffering the received encoded data, and a packet processing unit for subjecting the encoded data to a packet process so as to generate bitstreams and providing the generated bitstreams to the image decoding device.

20   **[0090]**       Further, a display unit for displaying an image decoded by the image decoding device may be added to configure a display device. In that case, the display unit reads a decoded image signal generated by the decoded image signal superimposition unit 205 and stored in the decoded image memory 206 and displays the image on the screen.

**[0091]** Further, an imaging device may be configured by adding an imaging unit and feeding a captured image to the image encoding device. In that case, the imaging unit feeds an image signal captured to the block segmentation unit 101.

5 **[0092]** The above-described processes related to encoding and decoding can of course be implemented by hardware-based apparatus for transmission, storage, or reception. Alternatively, the processes can be implemented by firmware stored in a read-only memory (ROM), a flash memory, etc., or  
10 by software on a computer, etc. The firmware program or the software program may be made available on, for example, a computer readable recording medium. Alternatively, the programs may be made available from a server via a wired or wireless network. Still alternatively, the programs may be  
15 made available in the form of data transmission over terrestrial or satellite digital broadcast systems.

**[0093]** Described above is an explanation based on an exemplary embodiment. The embodiment is intended to be illustrative only and it will be understood by those skilled  
20 in the art that various modifications to constituting elements and processes could be developed and that such modifications are also within the scope of the present invention.

[DESCRIPTION OF THE REFERENCE NUMERALS]

25 **[0094]** 100 image encoding device, 101 block segmentation unit, 102 predicted image generation unit, 103 residual

signal generation unit, 104 orthogonal transform/quantization unit, 105 bitstream generation unit, 106 inverse-quantization/inverse-orthogonal transform unit, 107 decoded image signal superimposition unit, 108 decoded image memory, 5 200 image decoding device, 201 bitstream decoding unit, 202 block segmentation unit, 203 inverse-quantization/inverse-orthogonal transform unit, 204 predicted image generation unit, 205 decoded image signal superimposition unit, and 206 decoded image memory.

10 [INDUSTRIAL APPLICABILITY]

**[0095]** The present invention can be used for a technology of segmenting an image into blocks and encoding and decoding the image in units of blocks resulting from segmenting the image.

## [CLAIMS]

1. An image encoding device adapted to segment an image into blocks and encode the image in units of blocks resulting from segmenting the image, comprising:

a block segmentation unit that recursively segments the image into rectangles of a predetermined size to generate a block subject to encoding; and

an encoding unit that encodes block segmentation information of the block subject to encoding, wherein

the block segmentation unit includes:

a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when previous recursive segmentation is halving, the halving unit prohibits a target block subject to current recursive segmentation from being segmented in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

2. The encoding device according to claim 1, wherein

when the previous recursive segmentation is halving and the target block is of a predetermined size or smaller, the

halving unit prohibits the target block subject to the current recursive segmentation from being segmented in the same direction as the direction in which the block was segmented in the previous recursive segmentation.

3. The image encoding device according to claim 1, wherein

when the previous recursive segmentation is halving and the direction in which the block was segmented in the previous recursive segmentation is the vertical direction, the halving unit prohibits the target block subject to the current recursive segmentation from being segmented in the same direction as the direction in which the block was segmented in the previous recursive segmentation.

4. An image encoding device adapted to segment an image into blocks and encode the image in units of blocks resulting from segmenting the image, comprising:

a block segmentation unit that recursively segments the image into rectangles of a predetermined size to generate a block subject to encoding; and

an encoding unit that encodes block segmentation information of the block subject to encoding, wherein

the block segmentation unit includes:

a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a

vertical direction to generate four blocks; and

a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when the last but one recursive segmentation is halving and previous recursive segmentation is quartering, the block segmentation unit prohibits further segmentation of the target block.

5. An image encoding method adapted to segment an image into blocks and encode the image in units of blocks resulting from segmenting the image, comprising:

recursively segmenting the image into rectangles of a predetermined size to generate a block subject to encoding; and

encoding block segmentation information of the block subject to encoding, wherein

the segmenting includes:

quartering a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

halving a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when previous recursive segmentation is halving, the halving prohibits a target block subject to current recursive

segmentation from being segmented in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

6. An image encoding program adapted to segment an image into blocks and encode the image in units of blocks resulting from segmenting the image, the program comprising computer-implemented modules including:

a block segmentation module that recursively segments the image into rectangles of a predetermined size to generate a block subject to encoding; and

an encoding module that encodes block segmentation information of the block subject to encoding, wherein

the block segmentation module includes:

a quartering module that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

a halving module that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when previous recursive segmentation is halving, the halving module prohibits a target block subject to current recursive segmentation from being segmented in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

7. An image decoding device adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image, comprising:

a block segmentation unit that recursively segments the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and

a decoding unit that decodes the block subject to decoding, wherein

the block segmentation unit includes:

a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when previous recursive segmentation is halving, the decoding unit does not decode a flag indicating whether to segment a target block subject to current recursive segmentation in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

8. An image decoding device adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image, comprising:

a block segmentation unit that recursively segments the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and

a decoding unit that decodes the block subject to decoding, wherein

the block segmentation unit includes:

a quartering unit that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

a halving unit that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when the last but one recursive segmentation is halving and previous recursive segmentation is quartering, the decoding unit does not decode a flag indicating whether to segment the target block further.

9. An image decoding method adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image, comprising:

recursively segmenting the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and

decoding the block subject to decoding, wherein

the segmenting includes:

quartering a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

halving a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when previous recursive segmentation is halving, the decoding does not decode a flag indicating whether to segment a target block subject to current recursive segmentation in the same direction as a direction in which the block was segmented in the previous recursive segmentation.

10. An image decoding program adapted to segment an image into blocks and decode the image in units of blocks resulting from segmenting the image, the program comprising computer-implemented modules including:

a block segmentation module that recursively segments the image into rectangles of a predetermined size based on decoded block segmentation information to generate a block subject to decoding; and

a decoding module that decodes the block subject to decoding, wherein

the block segmentation module includes:

a quartering module that quarters a target block in recursive segmentation in a horizontal direction and a vertical direction to generate four blocks; and

a halving module that halves a target block in recursive segmentation in a horizontal direction or a vertical direction to generate two blocks, wherein

when previous recursive segmentation is halving, the decoding module does not decode a flag indicating whether to segment a target block subject to current recursive segmentation in the same direction as a direction in which the block was segmented in the previous recursive segmentation.