

(51) International Patent Classification:  
*H04N 7/32* (2006.01)(21) International Application Number:  
PCT/JP2011/066635(22) International Filing Date:  
14 July 2011 (14.07.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
12/837,438 15 July 2010 (15.07.2010) US(71) Applicant (for all designated States except US): **SHARP KABUSHIKI KAISHA** [JP/JP]; 22-22, Nagaïke-cho, Abeno-ku, Osaka-shi, Osaka, 5458522 (JP).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SEGALL, Christopher A.**, **ZHAO, Jie**.(74) Agent: **HARAKENZO WORLD PATENT & TRADE-MARK**; Daiwa Minamimorimachi Building, 2-6, Tenjinbashi 2-chome Kita, Kita-ku, Osaka-shi, Osaka, 5300041 (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, JP, 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, RO, RS, RU, 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.

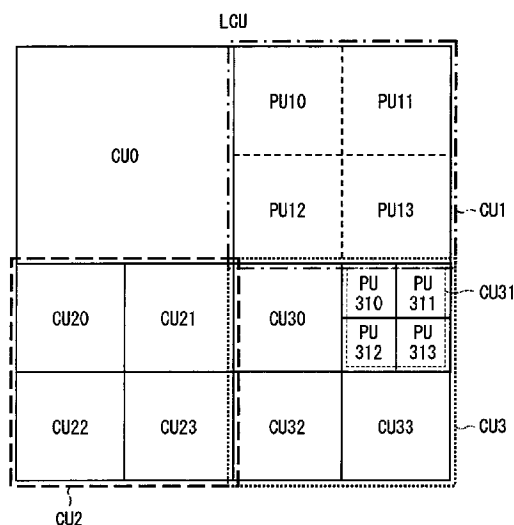
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, 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: PARALLEL VIDEO CODING BASED ON SCAN ORDER

FIG. 9B



(57) Abstract: Aspects of the present invention relate to methods for parallel video decoding. According to one aspect of the present invention, a first group of blocks is decoded using a first scan order and a second group of the blocks is decoded using a second scan order. The blocks within the first group may be predicted in parallel.

- 1 -

## DESCRIPTION

## TITLE OF INVENTION:

## PARALLEL VIDEO CODING BASED ON SCAN ORDER

## TECHNICAL FIELD

5           The present invention relates to a system for parallel video coding techniques.

## BACKGROUND ART

Existing video coding standards, such as H.264/AVC,  
10           generally provide relatively high coding efficiency at the expense of increased computational complexity. As the computational complexity increases, the encoding and/or decoding speeds tend to decrease. The use of parallel decoding and parallel encoding may improved the decoding  
15           and encoding speeds, respectively, particularly for multi-core processors. Also, parallel prediction patterns that depend solely on the number of prediction units within the block may be problematic for coding systems using other block structures because the number of prediction units may no  
20           longer correspond to the spatial size of the prediction unit.

## SUMMARY OF INVENTION

A preferred embodiment is a method for decoding video

- 2 -

comprising: (a) decoding a first block of video using a plurality of second blocks of said video; (b) decoding a first group of said second blocks using a first scan order in a manner such that each of said first group of said second blocks is predicted independently of the other ones of said second blocks not included within said first group; (c) decoding a second group of said second blocks using a second scan order in manner such that at least one block of said second group of said second blocks is predicted in a manner that is dependent on at least one block of said first group of said second blocks; (d) wherein said first and second scan orders are based upon whether said decoder is predicting said first group or said second group of said second blocks.

The foregoing and other objectives, features, and advantages of the invention will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

## BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 illustrates encoding patterns.

FIG. 2 illustrates prediction modes.

FIGS. 3A-3I illustrates intra-prediction modes.

FIG. 4 illustrates a 16 block macroblock with two partition groups.

FIGS. 5A-5D illustrate macroblocks with two partition groups.

FIGS. 6A-6B illustrate macroblocks with three partition groups.

5           FIG. 7 illustrates a macroblock with multiple partition groups.

FIG. 8 illustrates a coding unit split.

FIG. 9A illustrates spatial subdivision of a slice using various units and indices.

10           FIG. 9B illustrates spatial subdivisions of a largest coding unit suitable for intra-prediction.

FIG. 10 illustrates size based parallel decoding.

FIG. 11 illustrates one prediction unit with an intra\_split\_flag.

15           FIG. 12 illustrates type based parallel decoding.

FIG. 13 illustrates tree based parallel decoding.

FIG. 14A illustrates spatial windows based parallel decoding.

20           FIG. 14B illustrates the relationship between a window and a largest prediction unit.

FIG. 15 illustrates prediction direction in the angular mode of intra 8x8 macroblocks.

FIG. 16 illustrates arbitrary directional intra prediction modes defined by (dx, dy).

25           FIG. 17 illustrates pixels for combined intra prediction

and parallel intra prediction.

FIG. 18 illustrates block rotation.

## DESCRIPTION OF EMBODIMENTS

5           Intra-prediction based video encoding / decoding exploits spatial relationships within a frame, an image, or otherwise a block / group of pixels. At an encoder, a block of pixels may be predicted from neighboring previously encoded blocks of pixels, generally referred to as reconstructed blocks,  
10           typically located above and/or to the left of the current block, together with a prediction mode and a prediction residual for the block. A block may be any group of pixels that preferably shares the same prediction mode, the prediction parameters, the residual data and/or any other signaled data. At a  
15           decoder, a current block may be predicted, according to the prediction mode, from neighboring reconstructed blocks typically located above and/or to the left of the current block, together with the decoded prediction residual for the block. In many cases, the intra prediction uses, for example, 4x4,  
20           8x8, and 16x16 blocks of pixels.

          Referring to FIG. 1, with respect to the H.264/AVC video encoding standard, a 16x16 macroblock may include four 8x8 blocks or sixteen 4x4 blocks. The processing order for a group of four 8x8 blocks 2 of a 16x16 macroblock and for a  
25           group of sixteen 4x4 blocks 4 of a 16x16 macroblock may

- 5 -

have a zig-zag processing order, or any other suitable order. Typically, the current block within the macroblock being reconstructed is predicted using previously reconstructed neighboring blocks and/or macroblocks. Accordingly, the processing of one or more previous blocks of a 16x16 macroblock is completed before other blocks may be reconstructed using its neighbors within the macroblock. The intra 4x4 prediction has more serial dependency in comparison to intra 8x8 and 16x16 prediction. This serial dependency may increase the number of operating cycles within a processor therefore slowing down the time to complete the intra prediction, and may result in an uneven throughput of different intra prediction types.

Referring to FIG. 2, in H.264/AVC, the intra 4x4 prediction and 8x8 prediction have nine prediction modes. Pixel values in the current block may be predicted from pixels values in a reconstructed upper and/or left neighboring block(s) relative to the current block. The direction of the arrow depicting a mode indicates the prediction direction for the mode. The center point 11 does not represent a direction so this point may be associated with a DC prediction mode, or otherwise referred to as "mode 2". A horizontal arrow 12 extending to the right from the center point 11 may represent a horizontal prediction mode, also referred to as "mode 1". A vertical arrow 13 extending down from the center point 11

may represent a vertical prediction mode, also referred to as “mode 0”. An arrow 14 extending from the center point 11 diagonally downward to the right at approximately a 45 degree angle from horizontal may represent a diagonal down-right (DDR) prediction mode, also referred to as “mode 4”. An arrow 15 extended from the center point 11 diagonally downward to the left at approximately a 45 degree angle from horizontal may represent a diagonal down-left (DDL) prediction mode, also referred to as “mode 3”. Both the DDR and DDL prediction modes may be referred to as diagonal prediction modes. An arrow 16 extending from the center point 11 diagonally upward to the right at approximately a 22.5 degree angle from horizontal may represent a horizontal up (HU) prediction mode, also referred to as “mode 8”. An arrow 17 extending from the center point 11 diagonally downward to the right at approximately a 22.5 degree angle from horizontal may represent a horizontal down (HD) prediction mode, also referred to as “mode 6”. An arrow 18 extending from the center point 11 diagonally downward to the right at approximately a 67.5 degree angle from horizontal may represent a vertical down right (VR) prediction mode, also referred to as “mode 5”. An arrow 19 extending from the center point 11 diagonally downward to the left at approximately a 67.5 degree angle from horizontal may represent a vertical down left (VL) prediction mode, also

- 7 -

referred to as "mode 7". The HU, HD, VR, and VL prediction modes may be referred to collectively as intermediate angle prediction modes.

FIG. 3A illustrates an exemplary 4x4 block 20 of samples, labeled a-p that may be predicted from reconstructed, neighboring samples, labeled A-M. When samples are not available, such as for example when E-H are not available, they may be replaced by other suitable values.

Intra-prediction mode 0 (prediction mode direction indicated as 13 in FIG. 2) may be referred to as vertical mode intra prediction. In mode 0, or vertical mode intra prediction, the samples of a current block may be predicted in the vertical direction from the reconstructed samples in the block above the current block. In FIG. 3B, the samples labeled a-p in FIG. 3A are shown replaced with the label of the sample label from FIG. 3A from which they are predicted.

Intra-prediction mode 1 (prediction mode direction indicated as 12 in FIG. 2) may be referred to as horizontal mode intra prediction. In mode 1, or horizontal mode intra prediction, the samples of a block may be predicted in the horizontal direction from the reconstructed samples in the block to the left of the current block. FIG. 3C illustrates an exemplary horizontal prediction of the samples in a 4x4 block. In FIG. 3C, the samples labeled a-p in FIG. 3A are shown replaced with the label of the sample label from FIG. 3A from

which they are predicted.

Intra-prediction mode 3 (prediction mode direction indicated as 15 in FIG. 2) may be referred to as diagonal down left mode intra prediction. In mode 3, the samples of a block  
5 may be predicted from neighboring blocks in the direction shown in FIG. 3D.

Intra-prediction mode 4 (prediction mode direction indicated as 14 in FIG. 2) may be referred to as diagonal down right mode intra prediction. In mode 4, the samples of a  
10 block may be predicted from neighboring blocks in the direction shown in FIG. 3E.

Intra-prediction mode 5 (prediction mode direction indicated as 18 in FIG. 2) may be referred to as vertical right mode intra prediction. In mode 5, the samples of a block may  
15 be predicted from neighboring blocks in the direction shown in FIG. 3F.

Intra-prediction mode 6 (prediction mode direction indicated as 17 in FIG. 2) may be referred to as horizontal down mode intra prediction. In mode 6, the samples of a  
20 block may be predicted from neighboring blocks in the direction shown in FIG. 3G.

Intra-prediction mode 7 (prediction mode direction indicated as 19 in FIG. 2) may be referred to as vertical left mode intra prediction. In mode 7, the samples of a block may  
25 be predicted from neighboring blocks in the direction shown

in FIG. 3H.

Intra-prediction mode 8 (prediction mode direction indicated as 16 in FIG. 2) may be referred to as horizontal up mode intra prediction. In mode 8, the samples of a block may be predicted from neighboring blocks in the direction shown in FIG. 3I.

In intra-prediction mode 2, which may be referred to as DC mode, all samples labeled a-p in FIG. 3A may be replaced with the average of the samples labeled A-D and I-L in FIG. 3A.

The system may likewise support four 16x16 intra prediction modes in which the 16x16 samples of the macroblock are extrapolated from the upper and/or left hand encoded and reconstructed samples adjacent to the macroblock. The samples may be extrapolated vertically, mode 0 (similar to mode 0 for the 4x4 size block), or the samples may be extrapolated horizontally, mode 1 (similar to mode 1 for the 4x4 size block). The samples may be replaced by the mean, mode 2 (similar to the DC mode for the 4x4 size block), or a mode 3, referred to as plane mode, may be used in which a linear plane function is fitted to the upper and left hand samples.

In order to decrease the processing delays, especially when using parallel processors, it is desirable to process selected blocks of pixels of a larger group of pixels, such as a

- 10 -

macroblock, in a parallel fashion. A first group of blocks of pixels may be selected from a macroblock (or other larger set of pixels) and a second group of blocks of pixels may be selected from the remaining pixels of the macroblock.

5 Additional or alternative groups of blocks of pixels may be selected, as desired. A block of pixels may be any size, such as an  $m \times n$  size block of pixels, where  $m$  and  $n$  may be any suitable number. Preferably, each of the blocks within the first plurality of blocks are encoded using reconstructed pixel  
10 values from only one or more previously encoded neighboring macroblocks, and each of the blocks within the second plurality of blocks may be encoded using the reconstructed pixel values from previously encoded macroblocks and/or blocks associated with the first plurality of blocks. In this  
15 manner, the blocks within the first plurality of blocks may be decoded using reconstructed pixel values from only neighboring macroblocks, and then the blocks within the second plurality of blocks may be decoded using the reconstructed pixel values from reconstructed blocks  
20 associated with the first plurality of blocks and/or neighboring macroblocks. The encoding and decoding of one or more blocks may be, fully or partially, done in a parallel fashion.

For example, a macroblock with  $N$  blocks, the degree of  
25 parallelism may be  $N/2$ . The increased speed of  $4 \times 4$  intra

- 11 -

prediction for a 16x16 macroblock may be generally around a factor of 8, which is significant. Referring to FIG. 4, a macroblock has a size of MxN, where M and N may be any suitable number. The sixteen blocks 41-56 may be grouped into two (or more) sets of eight blocks (or otherwise) each according to a check board pattern (or other pattern). Eight blocks in a first set are shown as 42, 43, 46, 47, 50, 51, 54, and 55, and the eight blocks shown in the other set are 41, 44, 45, 48, 49, 52, 53, and 56. The first set of blocks may be decoded, or encoded, in parallel using previously reconstructed macroblocks, and then the second set of blocks may be decoded, or encoded, in parallel using the reconstructed blocks associated with the first set and/or previously reconstructed macroblocks. In some cases, the second set of blocks may start being decoded before the first set of blocks are completely decoded.

Alternative partition examples are shown in FIGS. 5A-5D. Referring to FIG. 5A, blocks 61-76 may be grouped in two groups. The first group may include 65-68 and 73-76, while the second group may include 61-64 and 69-72. Referring to FIG. 5B, blocks 81-96 may be grouped in two groups. The first group may include 82, 83, 85, 88, 89, 92, 94, and 95, while the second group may include 81, 84, 86, 87, 90, 91, 93, and 96. Referring to FIG. 5C, blocks 101-116 may be grouped in two groups. The first group may include 109-116, while

- 12 -

the second group may include 101-108. Referring to FIG. 5D, blocks 121-136 may be grouped in two groups. The first group may include 122, 124, 126, 128, 130, 132, 134, and 136, while the second group may include 121, 123, 125, 127, 129, 131, 133, and 135.

Alternatively, the macroblock may be partitioned into a greater number of partitions, such as three sets of blocks. Moreover, the partitions may have a different number of blocks. Further, the blocks may be the same or different sizes. In general, a first plurality of blocks may be predicted in the encoding process using reconstructed pixel values from only previously encoded neighboring macroblocks. A second plurality of blocks may be subsequently predicted in the encoding process using reconstructed pixel values from the previously encoded blocks associated with the first plurality of blocks and/or using reconstructed pixel values from previously encoded neighboring macroblocks. The third plurality of blocks may be subsequently predicted in the encoding process using reconstructed pixel values from the previously encoded blocks associated with the first plurality of blocks, and/or reconstructed pixel values from the previously encoded blocks associated with the second plurality of blocks, and/or reconstructed pixel values from previously encoded neighboring macroblocks. FIGS. 6A and 6B depict exemplary three-group partitions of a 16x16

- 13 -

macroblock. Referring to FIG. 6A, the first set includes eight blocks. The second set and the third set each include four blocks. Referring to FIG. 6B, the first set includes six blocks. The second set and the third set each include five blocks. In the case shown in FIG. 6A and 6B, for example, the first set of blocks may be predicted in the encoding process using reconstructed pixel values from only previously encoded neighboring macroblocks. Then the second set of blocks may be subsequently predicted in the encoding process using reconstructed pixel values from the previously encoded blocks associated with the first set of blocks and/or using reconstructed pixel values from previously encoded neighboring macroblocks. Then the third set of blocks may be subsequently predicted in the encoding process using reconstructed pixel values from the previously encoded blocks associated with the first set of blocks, and/or reconstructed pixel values from the previously encoded blocks associated with the second set of blocks, and/or reconstructed pixel values from previously encoded neighboring macroblocks. FIG. 7 shows an exemplary partition of 4x4 blocks in a 32x32 macroblock. Referring to FIG. 7, blocks may be grouped into nine sets. Also in the case of FIG. 7, the first set, the second set, and the third set may be predicted in this order. Note that in the case of FIG. 7, with an Nth set (where  $N \geq 4$ ), the Nth set is predicted with use of previously reconstructed

macroblocks, also previously reconstructed blocks, i.e. first set, ... to (N-1)th set.

The bit stream may require signaling which encoding pattern is used for the decoding, or otherwise the default decoding may be predefined.

In some embodiments, the neighboring upper and left macroblock pixel values may be weighted according to their distance to the block that is being predicted, or using any other suitable measure.

In some cases, the video encoding does not use fixed block sizes, but rather includes two or more different block sizes within a macroblock. In some implementations, the partitioning of an image may use the concepts of coding unit (CU), prediction unit (PU), and prediction partitions. At the highest level, this technique divides a picture into one or more slices. A slice is a sequence of largest coding units (LCU) that correspond to a spatial window within the picture. The coding unit, may be for example, a group of pixels containing one or more prediction modes / partitions and it may have residual data. The prediction unit, may be for example, a group of pixels that are predicted using the same prediction type, such as intra prediction or intra frame prediction. The prediction partition, may be for example, a group of pixels predicted using the same prediction type and prediction parameters. The largest coding unit, may be for

- 15 -

example, a maximum number of pixels for a coding unit. For example, a 64x64 group of pixels may correspond to a largest coding unit. These largest coding units are optionally subdivided to adapt to the underlying image content (and achieve efficient compression). This division is determined by an encoder and signaled to the decoder, and it may result in a quad-tree segmentation of the largest coding unit. The resulting partitions are called coding units, and these coding units may also be subsequently split. Coding unit of size CuSize may be split into four smaller coding units, CU0, CU1, CU2 and CU3 of size  $\text{CuSize} / 2$  as shown in FIG. 8. This is accomplished by signaling a `split_coding_unit_flag` to specify whether a coding unit is split into coding units with half horizontal and vertical size. The sub-division is recursive and results in a highly flexible partitioning approach.

Once no further splitting of the coding unit is signaled, the coding units are considered as prediction units. Each prediction unit may have multiple prediction partitions. For an intra coded prediction unit, this may be accomplished by signaling an `intra_split_flag` to specify whether a prediction unit is split into four prediction units with half horizontal and vertical size. Additional partitioning mechanisms may be used for inter-coded blocks, as desired. FIG. 9A illustrates an example spatial subdivision of one slice with various units and their indices (for Inter-prediction). Referring FIG. 9A, the

- 16 -

LARGEST\_CU is divided into four coding units CU0, CU1, CU2, and CU3. Further, each of the CU1, CU2, and CU3 are divided into four coding units. The coding units achieved by the sub-division are considered as prediction units. For example, the coding units achieved by the sub-division of CU1 are considered as PU10, PU11, PU12 and PU13. In addition, PU11 and PU12 have multiple partitions. PU11 has PP110, PP111, PP112, and PP113, and PU12 has PP120, PP121, PP122, and PP123. FIG. 9B illustrates spatial subdivisions of a largest coding unit suitable for intra-prediction. Referring FIG. 9B, LCU has CU0, CU1, CU2, and CU3. CU1, CU2, and CU3 include multiple prediction units. In this case, the processing for multiple coding units are preferably done in parallel. In addition, the processing for multiple prediction units are preferably done in parallel, such as CU20, CU21, CU22, CU23, of CU2; and such as the 4 divisions (PU10, PU11, PU12, and PU13) of CU1.

With the additional capability of using such flexible block structures, where the number of prediction units no longer corresponds to the spatial size of the prediction unit, it was determined that limitations should be placed on whether such parallel encoding and/or parallel decoding mode should be used. For relatively large prediction partitions there does not tend to be a significant increase in parallelism from otherwise processing multiple prediction partitions

sequentially. In addition, with different sized prediction units it would otherwise introduce significant computational complexity in order to accommodate a largest prediction (e.g., coding) unit with multiple different sized prediction units. Accordingly, it is desirable to only uses parallel encoding and/or decoding when the size of the blocks is less than a threshold size.

Referring to FIG. 10, preferably the system uses parallel intra prediction only for prediction units of the largest prediction unit that all contain partitions having the same size. The largest prediction unit, may be for example, the largest group of pixels being defined by a single set of data. This may be determined by inspection of the largest prediction unit, or other set of prediction units (S101). That may be signaled from within the bitstream by a flag, such as an `intra_split_flag`, for the prediction unit. When the `intra_split_flag` signals that the prediction unit is sub-divided into equally sized prediction partitions (YES in S102), then the parallel intra prediction system (a second technique) may be applied within that prediction unit (S104). When the `intra_split_flag` does not signal that the prediction unit is sub-divided into equally sized prediction partitions (NO in S102), then the parallel intra prediction system is preferably not applied (S103). In S103, The default decoding (a first technique) is performed. An exemplary splitting of the

- 18 -

prediction unit into four prediction partitions is illustrated in FIG. 11, which are then grouped into two sets for parallel processing. For example, partitions 1 and 2 may be grouped to one set and partitions 0 and 3 may be grouped to another set. The first set is then predicted using the prediction unit neighbors while the second set is predicted using prediction unit neighbors as well as the neighbors in the first set.

Referring to FIG. 12, in addition to the partitions having the same size, the system may further use parallel intra prediction (a second technique) (S125) across multiple prediction units that have prediction partitions that are of the same size and/or coding type (e.g., intra-coded vs. motion compensated). More specifically, the largest prediction unit, may be for example, the largest group of pixels being defined by a single set of data. This may be determined by inspection of the largest prediction unit, or other set of prediction units (S121). When all prediction partitions have same size (YES in S122) and the same coding type (YES in S123), then the parallel intra prediction system (the second technique) may be applied within that prediction unit (S125). When all prediction partitions does not have same size (NO in S122) or the same coding type (NO in S123), then the default decoding (a first technique) is performed (S124). Referring to FIG. 13, these prediction units preferably be spatially co-located within a coding unit that was subsequently split to create the

- 19 -

multiple prediction units (S131). Alternatively, the multiple prediction units may be spatially co-located within a coding unit that was recursively split to create the prediction units (S131). In other words, the prediction units have the same parent in the quad-tree. Since S132, S133, S134, and S135 are similar to S122, S123, S124 and S125 respectively, a description thereof is omitted here. Note that prediction partitions are originally of the same coding type. Accordingly, it is possible to omit the determination of the coding type, of S123 in Fig. 12 and S133 in Fig. 13.

In an embodiment the system may use parallel intra prediction across multiple coding units. The multiple coding units preferably have the same spatial size and prediction type (e.g., intra coded). Referring to FIG. 14A, in another embodiment, the parallel intra prediction technique may be based on the size of the prediction area. For example, the system may restrict the use of the parallel intra prediction technique to pixels within an NxN spatial window (S142). For example, the system may restrict use of the parallel intra prediction technique only to pixels within a 16x16 spatial window. Referring to FIG 14B, the spatial window is labeled as LPU (largest prediction unit) and includes data from a first coding unit, CU0, and a second coding unit, CU1. Note that the data used for processing the pixels within the window may be located outside of the window.

As described above, the spatial window may be referred to as a parallel unit. Alternatively, it may be referred to as a parallel prediction unit or parallel coding unit. The size of the parallel unit may be signaled in the bit-stream from an encoder to a decoder. Furthermore, it may be defined in a profile, defined in a level, transmitted as meta-data, or communicated in any other manner. The encoder may determine the size of the parallel coding unit and restricts the use of the parallel intra prediction technology to spatial pixels that do not exceed the size of the parallel unit. The size of the parallel unit may be signaled to the decoder. Additionally, the size of the parallel unit may be determined by table look, specified in a profile, specified in a level, determined from image analysis, determined by rate-distortion optimization, or any other suitable technique.

For a prediction partition that is intra-coded, the following technique may be used to reconstruct the block pixel values. First, a prediction mode is signaled from the encoder to the decoder. This prediction mode identifies a process to predict pixels in the current block from previously reconstructed pixel values. As a specific example, a horizontal predictor may be signaled that predicts a current pixel value from a previously reconstructed pixel value that is near and to the left of the current pixel location. As an alternative example, a vertical predictor may be signaled that

predicts a current pixel value from a previously reconstructed pixel value that is near and above the current pixel location. In general, pixel locations within a coding unit may have different predictions. The result is predicted pixel values for all the pixels of the coding unit.

Additionally, the encoder may send transform coefficient level values to the decoder. At the decoder, these transform coefficient level values are extracted from the bit-stream and converted to transform coefficients. The conversion may consist of a scaling operation, a table look-up operation, or any other suitable technique. Following the conversion, the transform coefficients are mapped into a two-dimensional transform coefficient matrix by a zig-zag scan operation, or other suitable mapping. The two-dimensional transform coefficient matrix is then mapped to reconstructed residual values by an inverse transform operation, or other suitable technique. The reconstructed residual values are added (or otherwise) to the predicted pixel values to form a reconstructed intra-predicted block.

The zig-zag scan operation and the inverse residual transform operation may depend on the prediction mode. For example, when a decoder receives a first prediction mode from an encoder for a first intra-predicted block, it uses the prediction process, zig-zag scan operation and inverse residual transform operation assigned to the first prediction

mode. Similarly, when a decoder receives a second prediction mode from an encoder for a second intra-predicted block, it uses the prediction process, zig-zag scan operation and inverse residual transform operation assigned to the second prediction mode. In general, the scan pattern used for encoding and decoding may be modified, as desired. In addition, the encoding efficiency may be improved by having the scan pattern further dependent on which group of the parallel encoding the prediction units or prediction partitions are part of.

In one embodiment the system may operate as follows: when a decoder receives a first prediction mode from an encoder for a first intra-predicted block that is assigned to a first partition, the decoder uses the prediction process, zig-zag scan operation and inverse residual transform operation assigned to the first prediction mode and the first partition. Similarly, when a decoder receives a second prediction mode from an encoder for a second intra-predicted block that is assigned to a second partition, the decoder uses the prediction process, zig-zag scan operation and inverse residual transform operation assigned the second prediction mode and the second partition. For example, the first and second partitions may correspond to a first and a second group for parallel encoding. Note that for the case that the first prediction mode and the second prediction mode have

- 23 -

the same value but the first partition and the second partition are not the same partition, then the first zig-zag scan operation (a first scan order) and first inverse residual transform operation may not be the same as the second zig-zag scan operation (a second scan order) and second inverse residual transform. This is true even if the first prediction process and second prediction process are the same. For example, the zig-zag scan operation for the first partition may use a horizontal transform and a vertical scan pattern, while the zig-zag scan operation for the second partition may use a vertical transform and a horizontal scan pattern.

There may be different intra prediction modes that are block size dependent. For block sizes of 8x8, 16x16, 32x32, there may be, for example, 33 intra prediction modes which provide substantially finer angle prediction compared to the 9 intra 4x4 prediction modes. While the 9 intra 4x4 prediction modes may be extended in some manner using some type of interpolation for finer angle prediction, this results in additional system complexity.

In the context of parallel encoding, including parallel encoding where the block sizes may have different sizes, the first set of blocks are generally predicted from adjacent macroblocks. Instead of extending the prediction modes of the 4x4 blocks (a first smaller block) to the larger blocks (e.g., 8x8, 16x16, 32x32, etc.) (a second larger block), thereby

increasing the complexity of the system, the system may reuse the existing prediction modes of the larger blocks. Therefore, the 4x4 block prediction modes may take advantage of the greater number of prediction modes identified for other sizes of blocks, such as those of 8x8, 16x16, and 32x32.

In many cases, the intra prediction modes of the 4x4 block size and prediction modes of the larger block sizes may be different. To accommodate the differences, it is desirable to map the 4x4 block prediction mode numbers to larger block prediction mode numbers. The mapping may be according to the prediction direction. For example, the intra prediction of a 4x4 block has 9 directional modes, while intra prediction of 8x8 block has 33 modes using angular prediction, intra prediction of block size 16x16 and 32x32 has 33 modes using arbitrary directional intra prediction (ADI). Angular prediction modes and the ADI prediction are shown in FIG. 15 and FIG. 16, respectively. FIG. 15 is a view illustrating an angular prediction mode having 33 types of available prediction directions. As illustrated in FIG. 15, the prediction directions correspond to prediction modes, respectively. The numbers in FIG. 15 represent the prediction modes; for instance, 0, 4, 5, 11, 12, 19, 20, 21, 22, 24, 33 represent predictions in horizontal directions of different angles. Moreover, 2 represents a DC prediction. The ADI is a

prediction method indicating the prediction directions by a coordinate of (dx,dy), as illustrated in FIG. 16. Even though the prediction modes of various blocks size may be different, for directional intra prediction, one mode may be mapped to another if they have the same or a close direction. For example, the system may map the value for mode 4 of the 4x4 block prediction to mode 19 of the 8x8 block prediction for the case that mode 4 related to a horizontal mode prediction and mode 9 related to a horizontal mode prediction.

To improve the prediction of a block the additional neighbors from the bottom and right may be used when available. Rather than extending the different prediction modes, the prediction from the bottom and the right neighbors may be done by rotating the block and then utilizing existing intra prediction modes. Predictions by two modes that are of 180 degree difference can be weighted interpolated as follows,

$$p(y,x) = w * p1(y,x) + (1-w)p2(y,x)$$

where p1 is the prediction that doesn't include the bottom and right neighbors (i.e. use top/left), and p2 is the prediction that doesn't include the above and left neighbors (i.e. use bottom/right), and w is a weighting factor. The weighting factor may be the weighted average process between the predictions from above and left neighbors ,and neighbors from bottom and right neighbors as follows:

- 26 -

First, derive value yTmp at pixel (x,y) as weighted average of p1 and p2, where weight is according to the distance to the above and bottom neighbors

$$yTmp = ( p1*(N-y) + p2*y ) / N ;$$

5        Second, derive value xTmp at pixel (x,y) as weighted average of p1 and p2, where weight is according to the distance to the left and right neighbors.

$$xTmp = ( p1* (N-x) + p2* x ) / N;$$

10        Namely, weight are proportional to the distance between each predicted pixels and the top/bottom and left/right neighbor. The N is the block width, p1 is the prediction that doesn't include the bottom and right neighbors, and p2 is the prediction that doesn't include the above and left neighbors.

15        Third, the final predicted value at pixel (x,y) is a weighted average of xTmp and yTmp. The weight depends on the prediction direction. For each direction, represent its angle as (dx, dy), as represented in ADI mode in FIG. 16. For mode without direction, it is preferable to set dx=1, dy=1.

$$20 \quad p(y,x) = ( abs(dx) * xTmp + abs(dy) * yTmp ) / (abs(dx)+abs(dy));$$

25        The encoder may make the decision on whether to perform weighted intra prediction or not, and signal the decision in the bitstream. A sample syntax for adding this weighted intra prediction flag is shown below below tables 1, 2, and 3. This may be signaled at coding unit level and/or

prediction unit level where the parallel intra prediction occurs.

(Table 1)

|                                      |   |            |
|--------------------------------------|---|------------|
| seq_parameter_set_rbsp( ) {          | C | Descriptor |
| profile_idc                          | 0 | u(8)       |
| reserved_zero_8bits /* equal to 0 */ | 0 | u(8)       |
| level_idc                            | 0 | u(8)       |
| seq_parameter_set_id                 | 0 | ue(v)      |
| bit_depth_luma_minus8                | 0 | ue(v)      |
| bit_depth_chroma_minus8              | 0 | ue(v)      |
| increased_bit_depth_luma             | 0 | ue(v)      |
| increased_bit_depth_chroma           | 0 | ue(v)      |
| log2_max_frame_num_minus4            | 0 | ue(v)      |
| log2_max_pic_order_cnt_lsb_minus4    | 0 | ue(v)      |
| max_num_ref_frames                   | 0 | ue(v)      |
| gaps_in_frame_num_value_allowed_flag | 0 | u(1)       |
| log2_min_coding_unit_size_minus3     | 0 | ue(v)      |
| max_coding_unit_hierarchy_depth      | 0 | ue(v)      |
| log2_min_transform_unit_size_minus2  | 0 | ue(v)      |
| max_transform_unit_hierarchy_depth   | 0 | ue(v)      |
| log2_parallel_unit_size              | 0 | ue(v)      |
| pic_width_in_luma_samples            | 0 | u(16)      |
| pic_height_in_luma_samples           | 0 | u(16)      |
| rbbsp_trailing_bits( )               | 0 |            |
| }                                    |   |            |

(Table 2)

|                                             |   |                 |
|---------------------------------------------|---|-----------------|
| coding_unit( x0, y0, currCodingUnitSize ) { | C | Descriptor      |
| ....                                        |   |                 |
| split_coding_unit_flag                      |   | (1)   ae(v)     |
| ...                                         |   |                 |
| alf_flag                                    | 2 | u(1)  <br>ae(v) |

- 28 -

|                                                                                      |              |                 |
|--------------------------------------------------------------------------------------|--------------|-----------------|
| if( split_coding_unit_flag ) {                                                       |              |                 |
| splitCodingUnitSize = currCodingUnitSize >><br>1                                     |              |                 |
| x1 = x0 + splitCodingUnitSize                                                        |              |                 |
| y1 = y0 + splitCodingUnitSize                                                        |              |                 |
| if (is_parallel_unit)                                                                | 2            | u(1)  <br>ae(v) |
| weighted_bipred_flag                                                                 |              |                 |
| coding_unit( x0, y0, splitCodingUnitSize )                                           | 2   3  <br>4 |                 |
| if( x1 < PicWidthInSamples <sub>L</sub> )                                            |              |                 |
| coding_unit( x1, y0,<br>splitCodingUnitSize )                                        | 2   3  <br>4 |                 |
| if( y1 < PicHeightInSamples <sub>L</sub> )                                           |              |                 |
| coding_unit( x0, y1,<br>splitCodingUnitSize )                                        | 2   3  <br>4 |                 |
| if( x1 < PicWidthInSamples <sub>L</sub> && y1 <<br>PicHeightInSamples <sub>L</sub> ) |              |                 |
| coding_unit( x1, y1,<br>splitCodingUnitSize )                                        | 2   3  <br>4 |                 |
| } else {                                                                             |              |                 |
| prediction_unit( x0, y0, currCodingUnitSize )                                        | 2            |                 |
| ....                                                                                 |              |                 |
| }                                                                                    |              |                 |
| }                                                                                    |              |                 |

(Table 3)

|                                               |   |                 |
|-----------------------------------------------|---|-----------------|
| prediction_unit( x0, y0, currPredUnitSize ) { | C | Descriptor      |
| .....                                         |   |                 |
| if( PredMode == MODE_INTRA ) {                |   |                 |
| planar_flag                                   | 2 | u(1)  <br>ae(v) |
| if(planar_flag) {                             |   |                 |
| ...                                           |   |                 |

- 29 -

|                                                          |   |                  |
|----------------------------------------------------------|---|------------------|
| }                                                        |   |                  |
| } else {                                                 |   |                  |
| if(entropy_coding_mode_flag)                             |   |                  |
| intra_split_flag                                         | 2 | ae(v)            |
| combined_intra_pred_flag                                 | 2 | u(1)  <br>ae(v)  |
| if (is_parallel_unit)                                    | 2 | u(1)  <br>ae(v)  |
| weighted_bipred_flag                                     |   |                  |
| for( i = 0; i <<br>( intra_split_flag ? 4 : 1 ); i++ ) { |   |                  |
| prev_intra_luma_pred_<br>flag                            | 2 | u(1)  <br>ae(v)  |
| if( !prev_intra_pred_luma_flag )                         |   |                  |
| rem_intra_luma_pred_mode                                 | 2 | ue(v)  <br>ae(v) |
| }                                                        |   |                  |
| }                                                        |   |                  |
| if( chroma_format_idc != 0 )                             |   |                  |
| intra_chroma_pred_mode                                   | 2 | ue(v)  <br>ae(v) |
| }                                                        |   |                  |
| }                                                        |   |                  |
| else if( PredMode == MODE_INTER ) {                      |   |                  |
| ...                                                      |   |                  |
| }                                                        |   |                  |
| }                                                        |   |                  |

The log2\_parallel\_unit\_size specifies the size of the parallel prediction unit. The variable ParallelUnitSize is derived as

- 30 -

ParallelUnitSize = MaxTransformUnitSize >>

log2\_parallel\_unit\_size,

where MaxTransformUnitSize is a largest value of the transform size. Semantics for the is\_parallel\_unit and  
5 weighted\_bipred\_flag may be defined as is\_parallel\_unit is true when currCodingUnitSize is less than or equal to ParallelUnitSize, where currCodingUnitSize is the size of current block and ParallelUnitSize is the size of the parallel prediction unit. weighted\_bipred\_flag equal to 1 defines the  
10 use of weighted bi-directional prediction for second pass units during intra-coding and equal to 0 defines the use of signal direction prediction for second pass units.

The intra prediction may be a weighted combination of an ADI prediction with a pixel-by-pixel mean prediction.  
15 Local mean is constructed as the average of reconstructed pixel values to the left, top-left, and above the current picture. While this is suitable for most of the image, it is problematic for the boundary pixels of the first set blocks of the parallel group, since such pixels may not be reconstructed.

20 By referring FIG. 17, the following describes pixels for combined intra prediction and parallel intra prediction. FIG. 17 illustrates pixels for combined intra prediction and parallel intra prediction. One technique to account for boundary issues is to use the parallel unit neighbors to

- 31 -

replace the unavailable pixels in the local mean calculation. For example, for the highlighted pixel in FIG. 17, the system may use AL' as the above left pixel, and use L' as the left pixel in the local mean calculation. These neighbors may be given  
5 different weights according to their distance to the pixel. In another embodiment, the system may use other available pixels in the adaptation, including those available from the ADI prediction and not yet processed by the combined intra prediction process. For example, for the highlighted pixel in  
10 FIG. 17, the system may include its above right pixel AR in the adaptation, and/or may also include its right pixel R, or bottom pixel B, which are already predicted by ADI prediction, but not yet going through the combined intra process. In one embodiment, the combined intra prediction may include  
15 modifying at the boundary of the pixels of the first set blocks in parallel intra prediction. In another embodiment, the combined intra prediction may be skipped for the boundary pixels of the first set blocks in parallel intra prediction.

Another technique for combined intra prediction with  
20 parallel intra prediction is to start the combined intra prediction from the bottom right pixel of a block if the right and bottom neighbors are available. This may be done by rotating the block, then performing the local mean adaptation and weighted average with utilizing existing intra prediction  
25 method such as the ADI or other types of intra prediction.

The rotation process is illustrated below in FIG. 18. In another embodiment, the result of combined intra prediction started from the upper left corner of a block, and the result of combined intra prediction started from the bottom right corner of a block can be weighted average together.

In WD3: Working Draft 3 of High-Efficiency Video Coding (HEVC), the partitioning of an image may use the concepts of coding unit (CU), prediction tree (PT), and prediction unit (PU). The macroblock in the foregoing embodiment corresponds to LCU (Largest Coding Unit; may also be called a root of a Coding Tree) as defined in WD3. Moreover, the macroblock and block in the embodiment correspond to the CU (Coding Unit; may also be called a leaf of the Coding Tree), the PU (Prediction Unit), or TU (Transformation Unit) in WD3.

The terms and expressions which have been employed in the foregoing specification are used therein as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being recognized that the scope of the invention is defined and limited only by the claims which follow.

## CLAIMS

1. A method for decoding video comprising:

(a) decoding a first block of video using a plurality of

5 second blocks of said video;

(b) decoding a first group of said second blocks

using a first scan order in a manner such that each of said

first group of said second blocks is predicted independently

of the other ones of said second blocks not included within

10 said first group;

(c) decoding a second group of said second blocks

using a second scan order in manner such that at least one

block of said second group of said second blocks is

predicted in a manner that is dependent on at least one

15 block of said first group of said second blocks;

(d) wherein said first and second scan orders are based upon whether said decoder is predicting said first group or said second group of said second blocks.

5           2. The method of claim 1 wherein said first group is predicted in a first manner, said second group is predicted in said first manner, and said first scan order is different than said second scan order.

10           3. The method of claim 1 wherein said scan order of a particular block having a particular prediction mode is dependent on whether it is a member of said first group or said second group.

15           4. The method of claim 1 wherein each of said second blocks of said video have the same size.

5. The method of claim 1 wherein each of said  
second blocks of said video are predicted using the same  
technique.

5

6. The method of claim 4 wherein said size is  
signaled from within the bitstream by a flag.

7. The method of claim 4 wherein said decoding is  
capable of selecting the size of said second blocks having a  
non-uniform size.

10

8. The method of claim 1 wherein a plurality of said  
first group of said second blocks are predicted in parallel.

15

9. The method of claim 8 wherein a plurality of said second group of said second blocks are predicted in parallel.

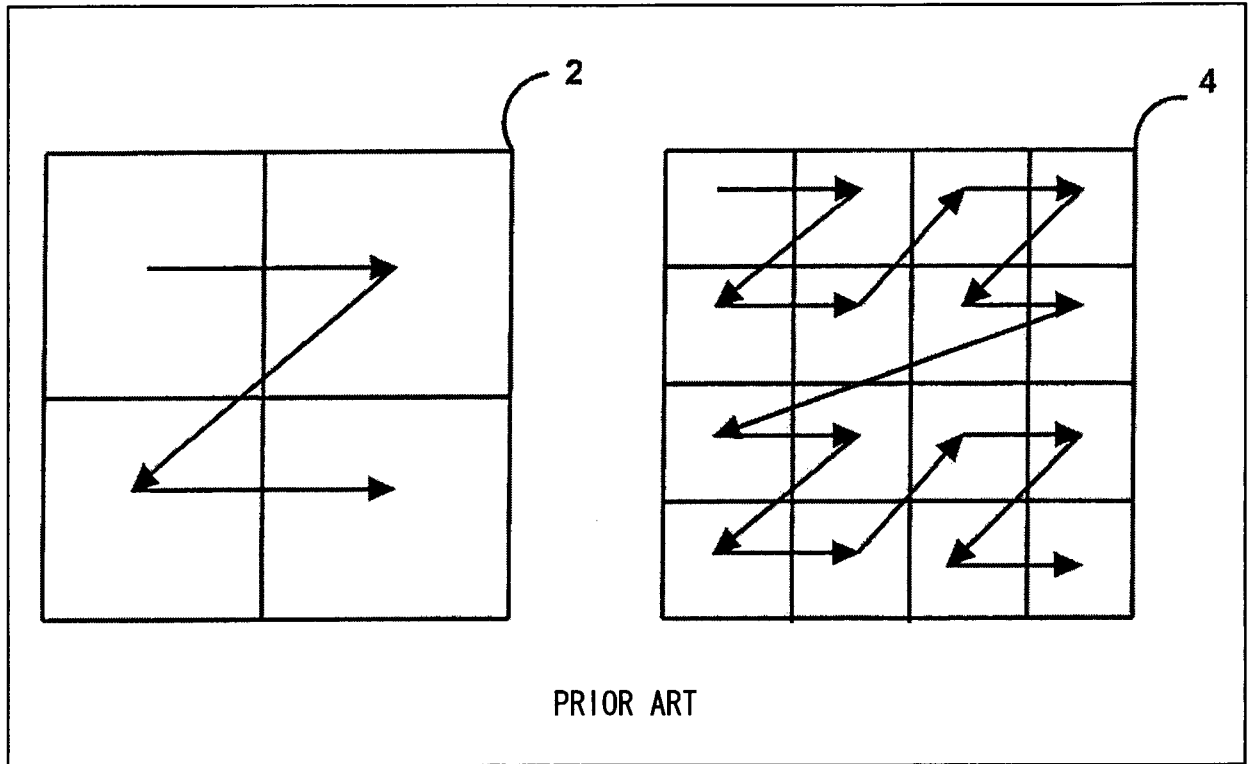
10. The method of claim 9 wherein said first group is  
5 said decoded prior to said second group being said decoded.

11. The method of claim 1 wherein said second blocks of said video have the same prediction type.

10 12. The method of claim 1 wherein the predicting of at least one of said first group of said second blocks is modified at a boundary of said at least one of said first group.

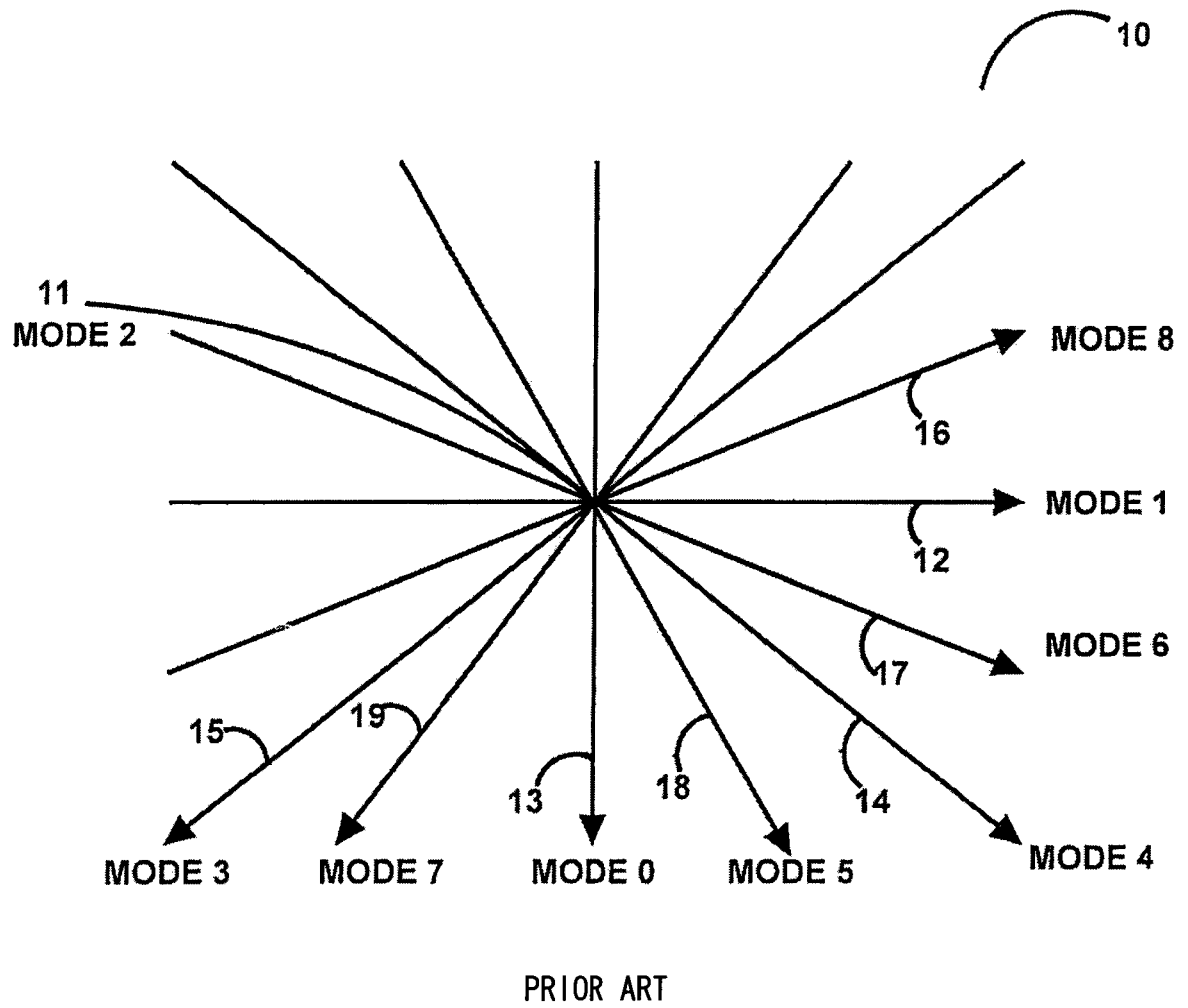
13. The method of claim 1 wherein said decoder is capable of selecting the size of said second blocks having a non-uniform size.

F I G. 1



2/24

FIG. 2



3/24

FIG. 3A

|   |   |   |   |   |                                                                                    |   |   |   |
|---|---|---|---|---|------------------------------------------------------------------------------------|---|---|---|
| M | A | B | C | D | E                                                                                  | F | G | H |
| I | a | b | c | d |  |   |   |   |
| J | e | f | g | h |                                                                                    |   |   |   |
| K | i | j | k | l |                                                                                    |   |   |   |
| L | m | n | o | p |                                                                                    |   |   |   |

PRIOR ART

FIG. 3B

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| M | A | B | C | D | E | F | G | H |
| I | A | B | C | D |   |   |   |   |
| J | A | B | C | D |   |   |   |   |
| K | A | B | C | D |   |   |   |   |
| L | A | B | C | D |   |   |   |   |

PRIOR ART

FIG. 3C

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| M | A | B | C | D | E | F | G | H |
| I | I | I | I | I |   |   |   |   |
| J | J | J | J | J |   |   |   |   |
| K | K | K | K | K |   |   |   |   |
| L | L | L | L | L |   |   |   |   |

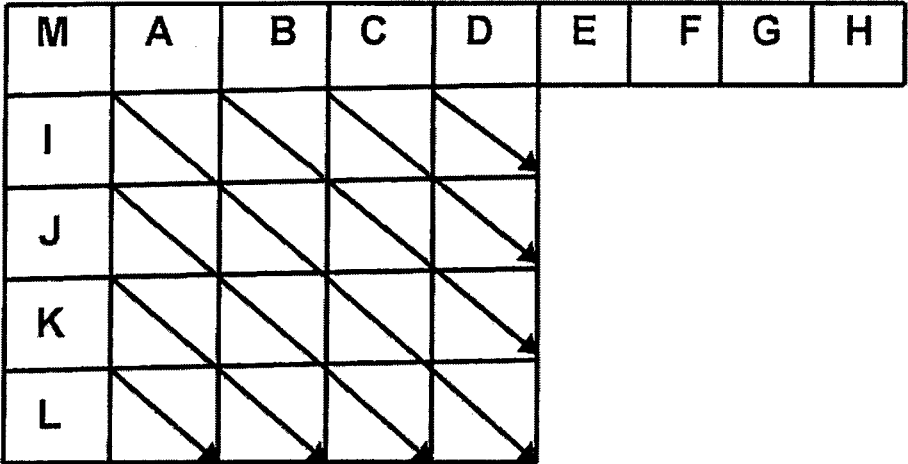
PRIOR ART

FIG. 3D

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| M | A | B | C | D | E | F | G | H |
| I |   |   |   |   |   |   |   |   |
| J |   |   |   |   |   |   |   |   |
| K |   |   |   |   |   |   |   |   |
| L |   |   |   |   |   |   |   |   |

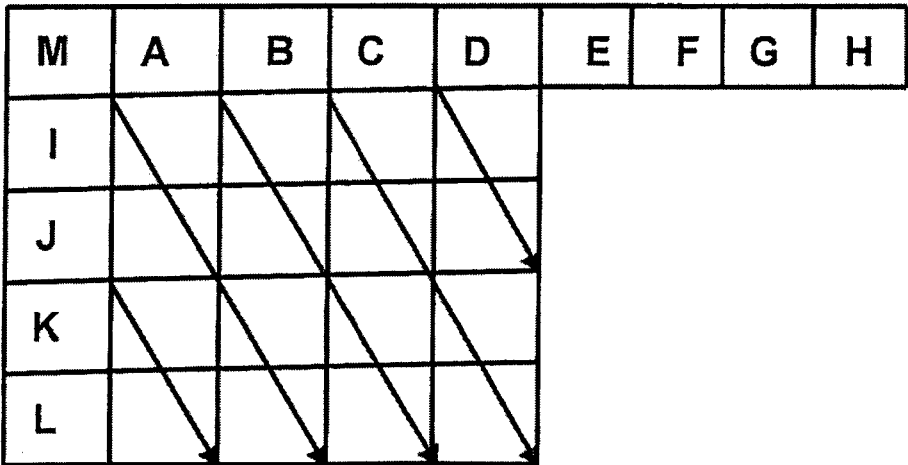
PRIOR ART

FIG. 3E



PRIOR ART

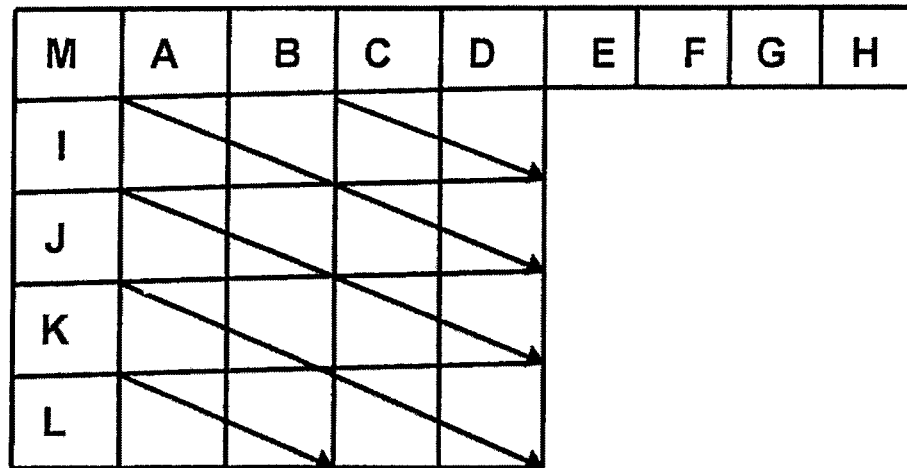
FIG. 3F



PRIOR ART

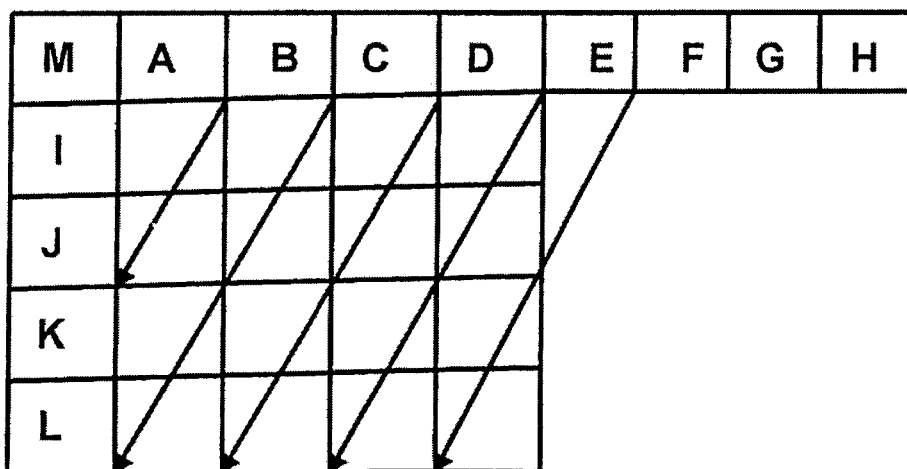
6/24

FIG. 3G



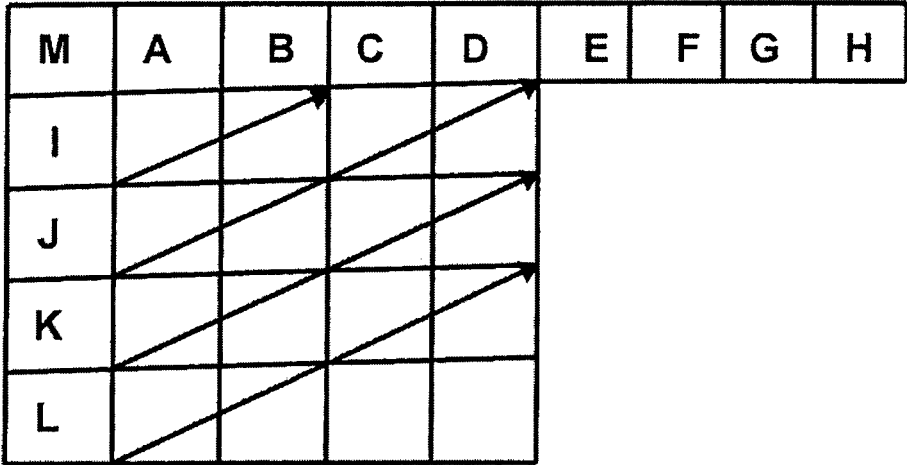
PRIOR ART

FIG. 3H



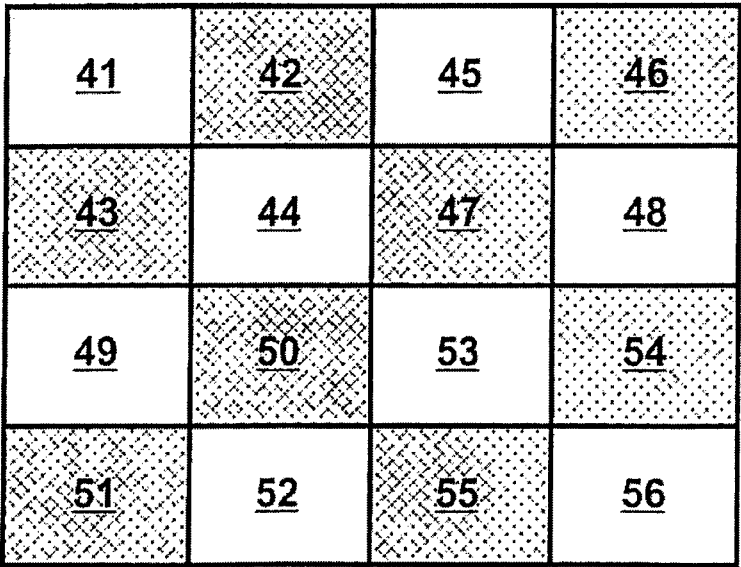
PRIOR ART

FIG. 3 I



PRIOR ART

FIG. 4



8/24

FIG. 5A

|           |           |           |           |
|-----------|-----------|-----------|-----------|
| <u>61</u> | <u>62</u> | <u>63</u> | <u>64</u> |
| <u>65</u> | <u>66</u> | <u>67</u> | <u>68</u> |
| <u>69</u> | <u>70</u> | <u>71</u> | <u>72</u> |
| <u>73</u> | <u>74</u> | <u>75</u> | <u>76</u> |

FIG. 5B

|           |           |           |           |
|-----------|-----------|-----------|-----------|
| <u>81</u> | <u>82</u> | <u>83</u> | <u>84</u> |
| <u>85</u> | <u>86</u> | <u>87</u> | <u>88</u> |
| <u>89</u> | <u>90</u> | <u>91</u> | <u>92</u> |
| <u>93</u> | <u>94</u> | <u>95</u> | <u>96</u> |

9/24

FIG. 5C

|            |            |            |            |
|------------|------------|------------|------------|
| <u>101</u> | <u>102</u> | <u>103</u> | <u>104</u> |
| <u>105</u> | <u>106</u> | <u>107</u> | <u>108</u> |
| <u>109</u> | <u>110</u> | <u>111</u> | <u>112</u> |
| <u>113</u> | <u>114</u> | <u>115</u> | <u>116</u> |

FIG. 5D

|            |            |            |            |
|------------|------------|------------|------------|
| <u>121</u> | <u>122</u> | <u>123</u> | <u>124</u> |
| <u>125</u> | <u>126</u> | <u>127</u> | <u>128</u> |
| <u>129</u> | <u>130</u> | <u>131</u> | <u>132</u> |
| <u>133</u> | <u>134</u> | <u>135</u> | <u>136</u> |

10/24

FIG. 6A

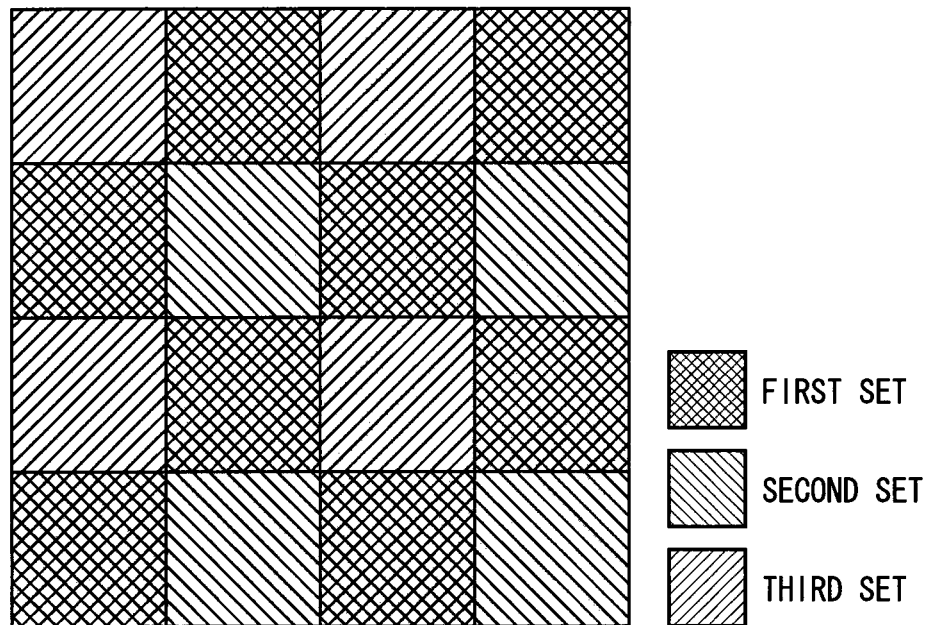
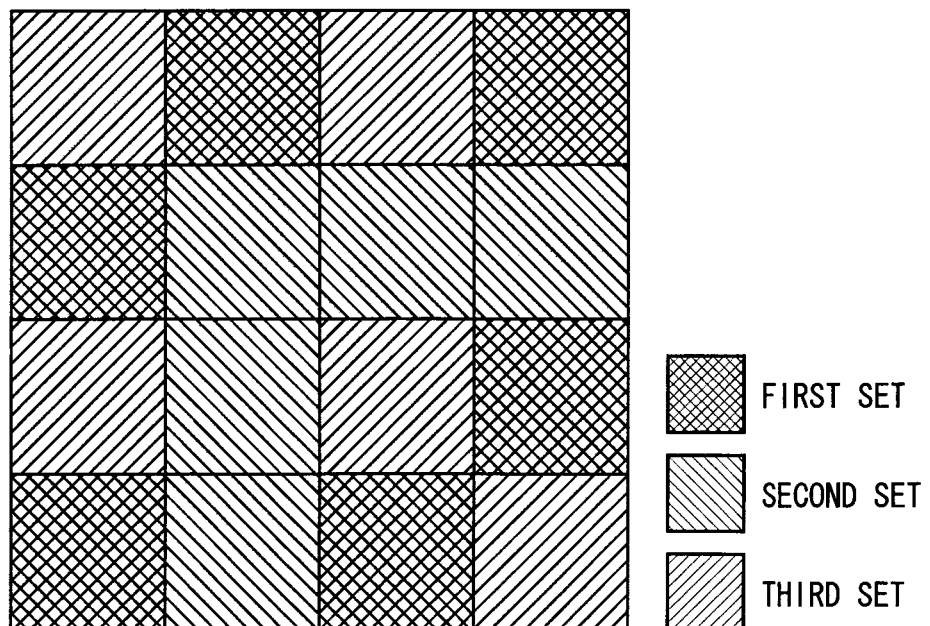
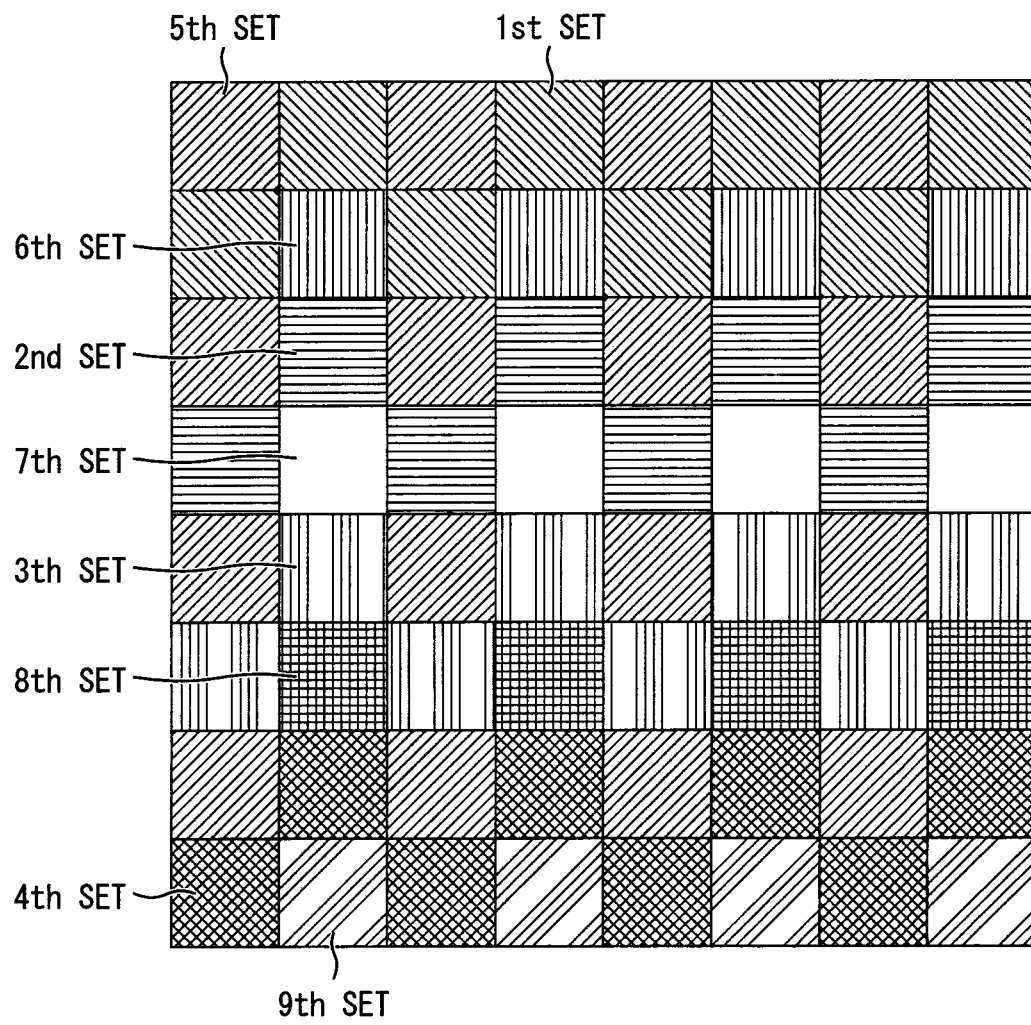


FIG. 6B



11/24

FIG. 7



F I G . 8

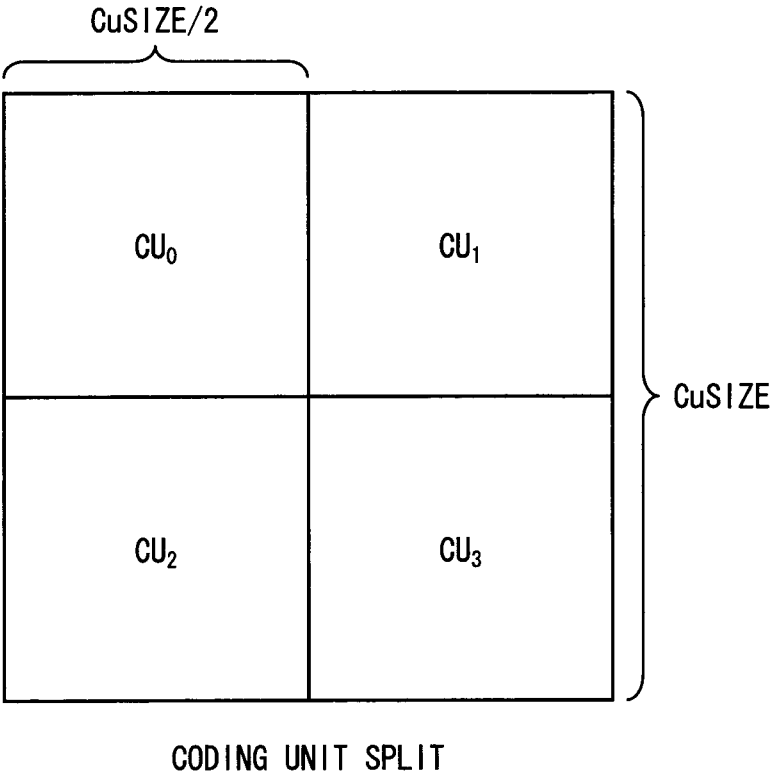
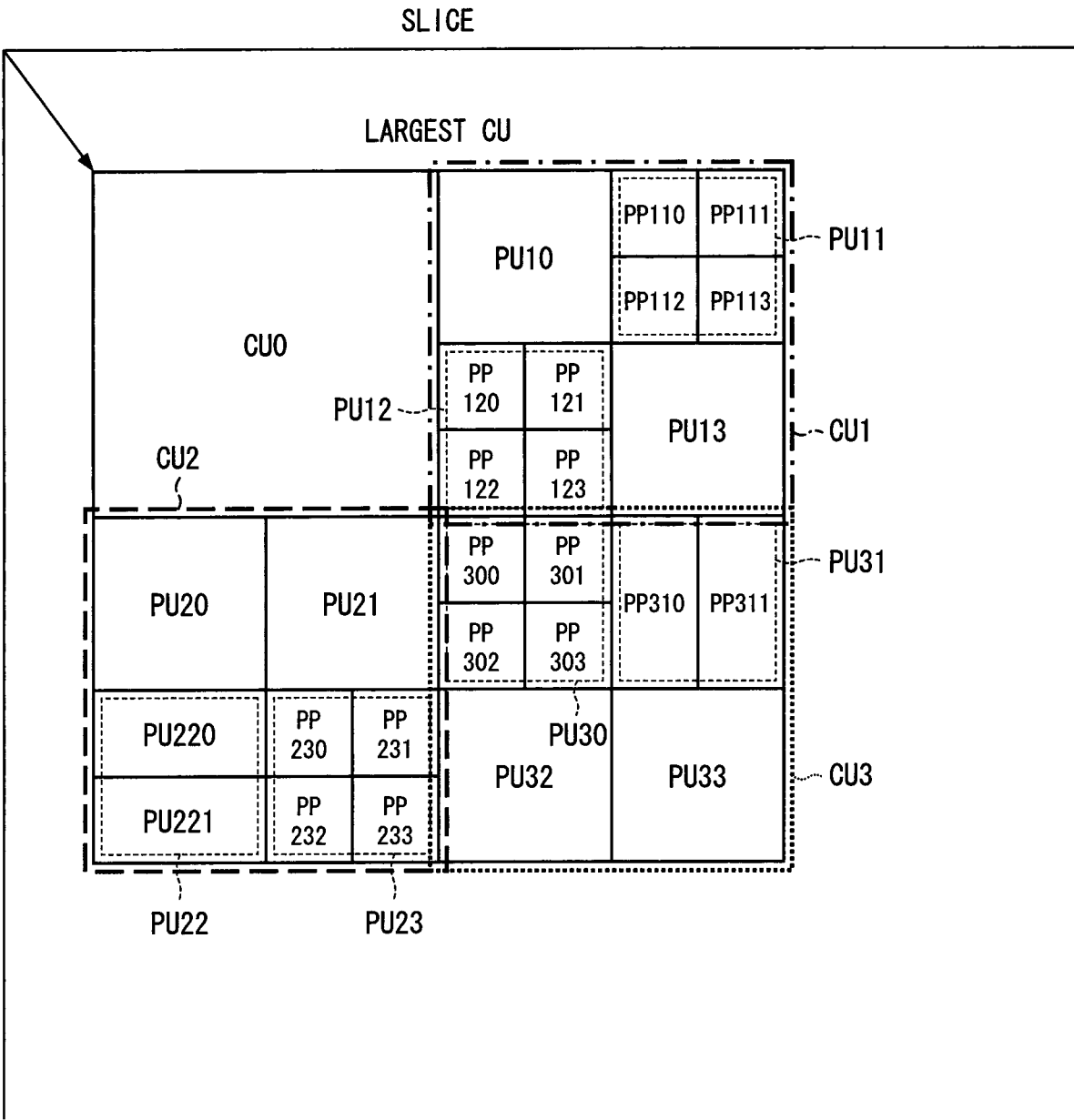
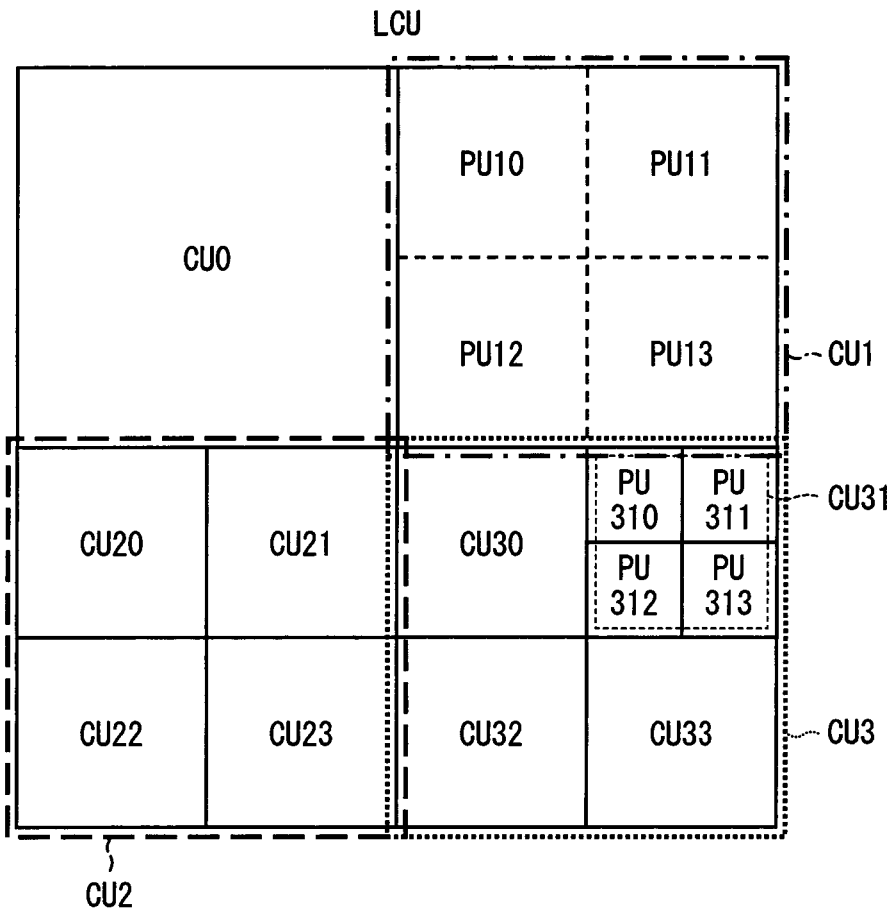


FIG. 9A

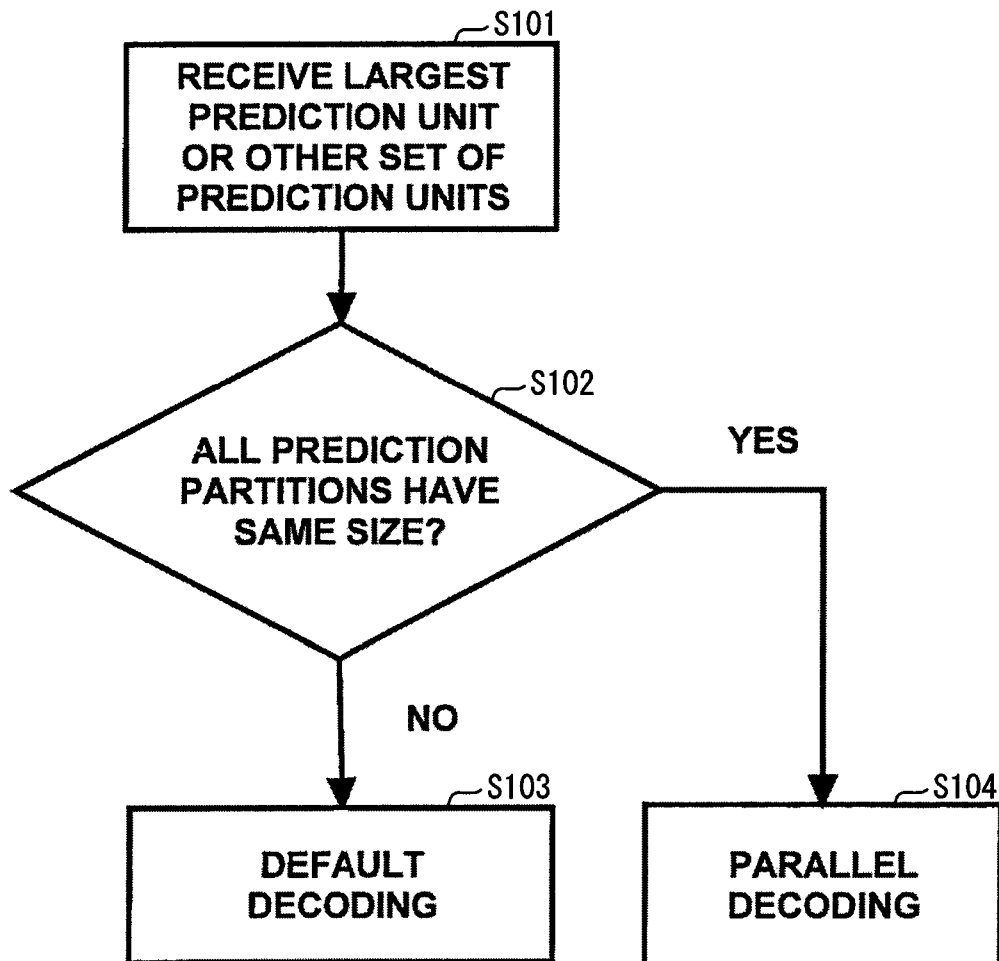


F I G . 9 B



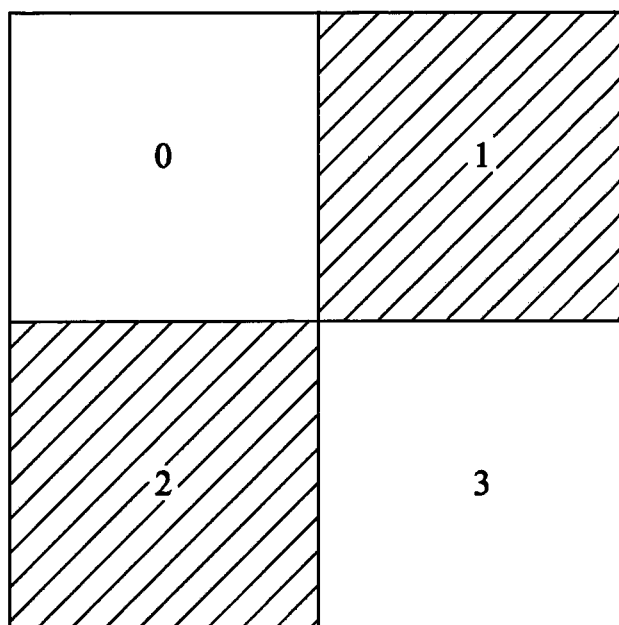
15/24

FIG. 10



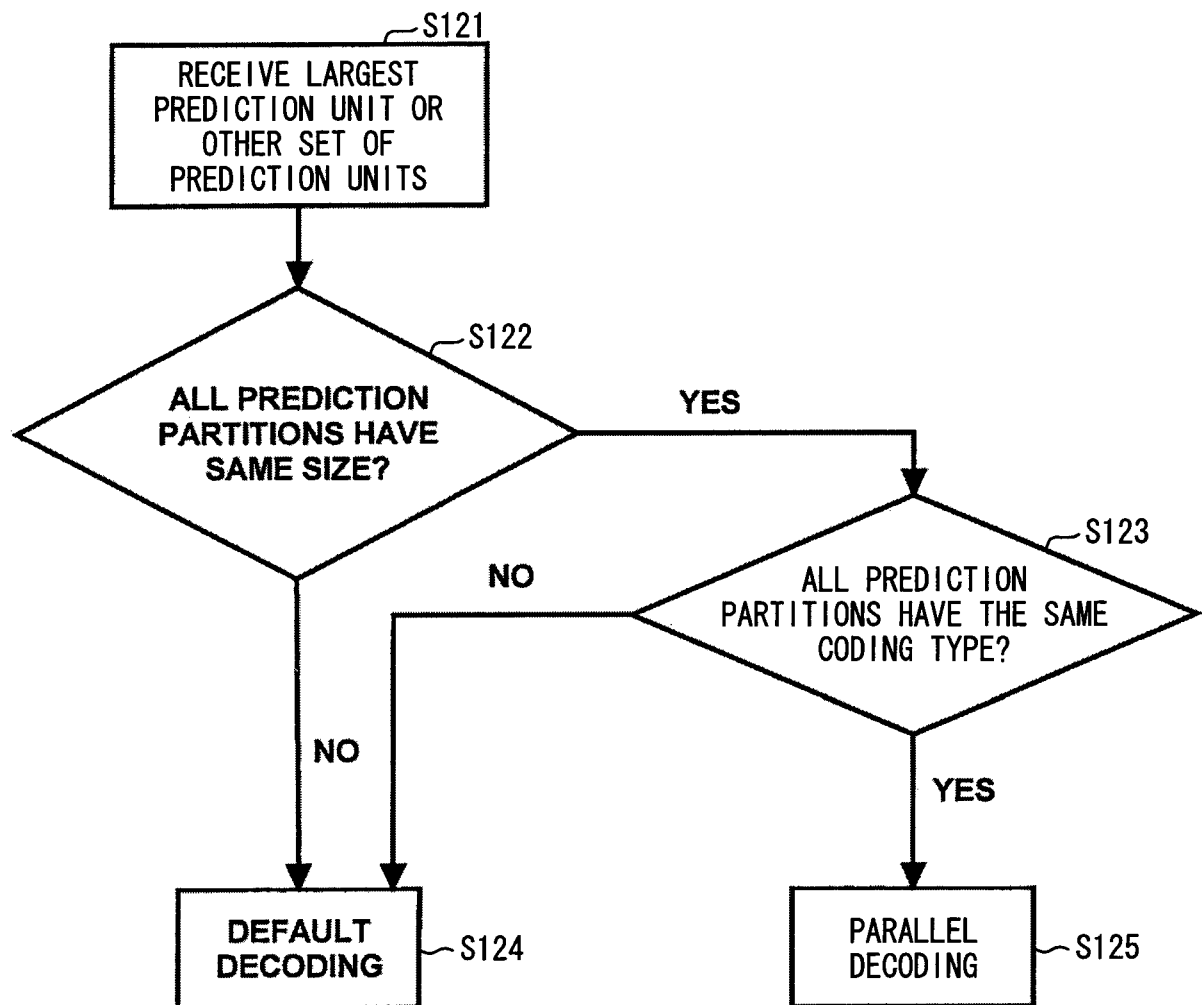
16/24

FIG. 11



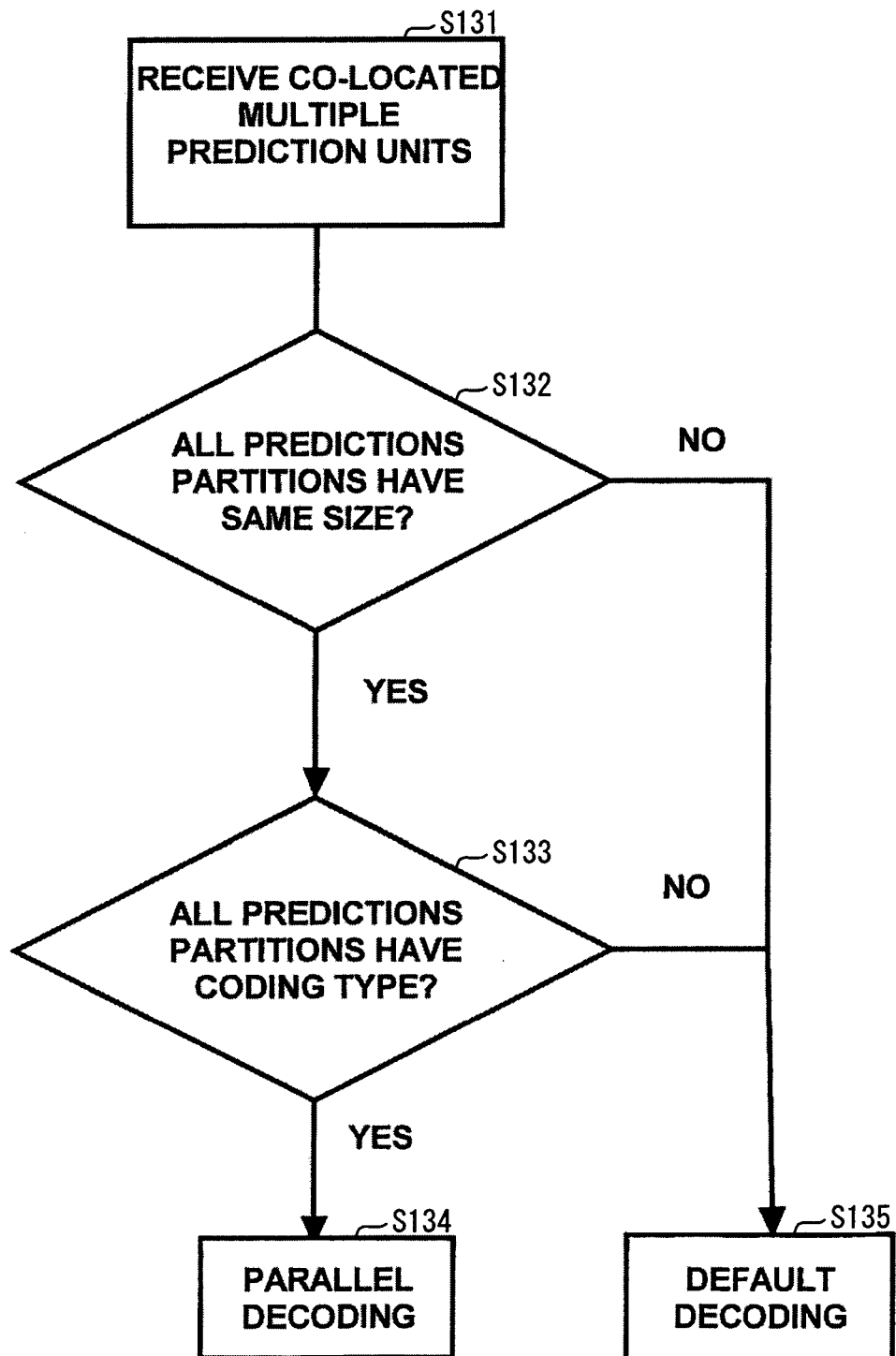
17/24

FIG. 12



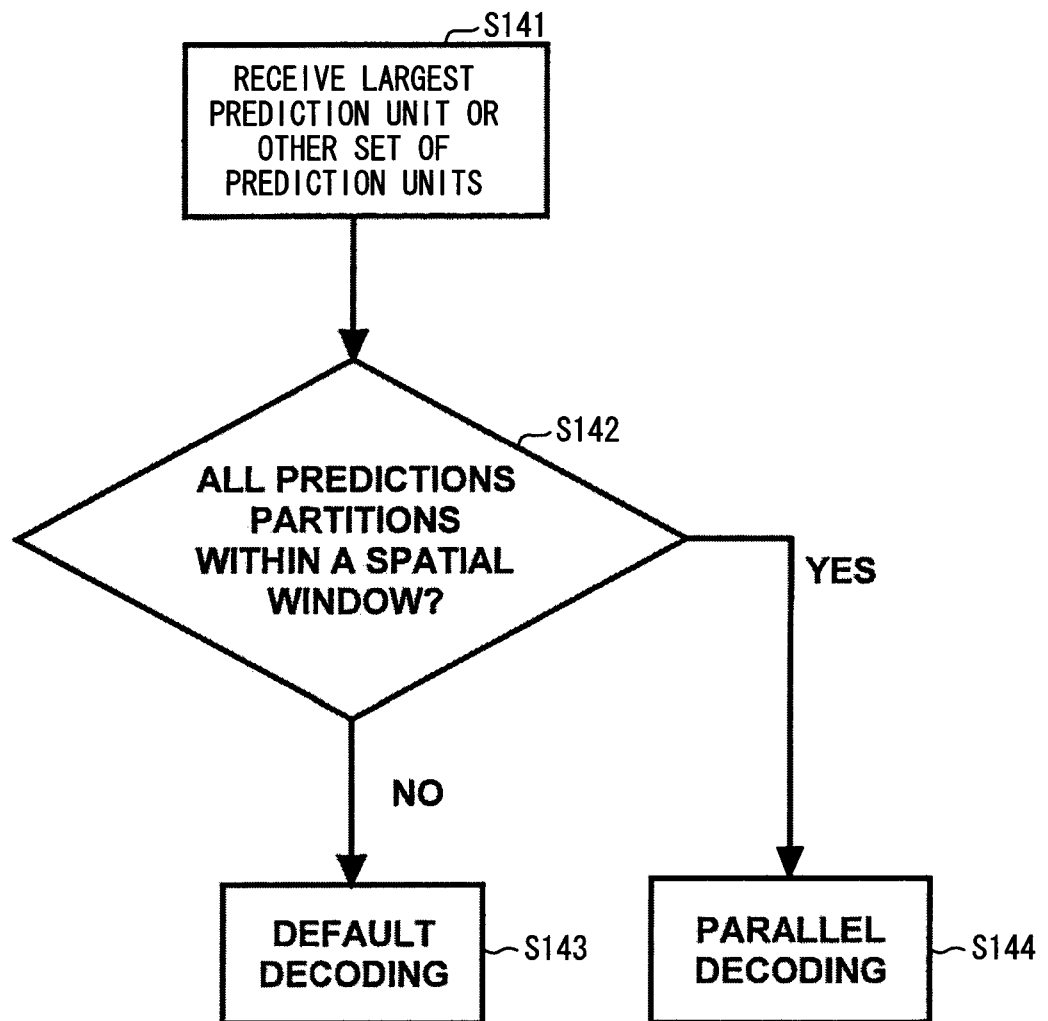
18/24

FIG. 13

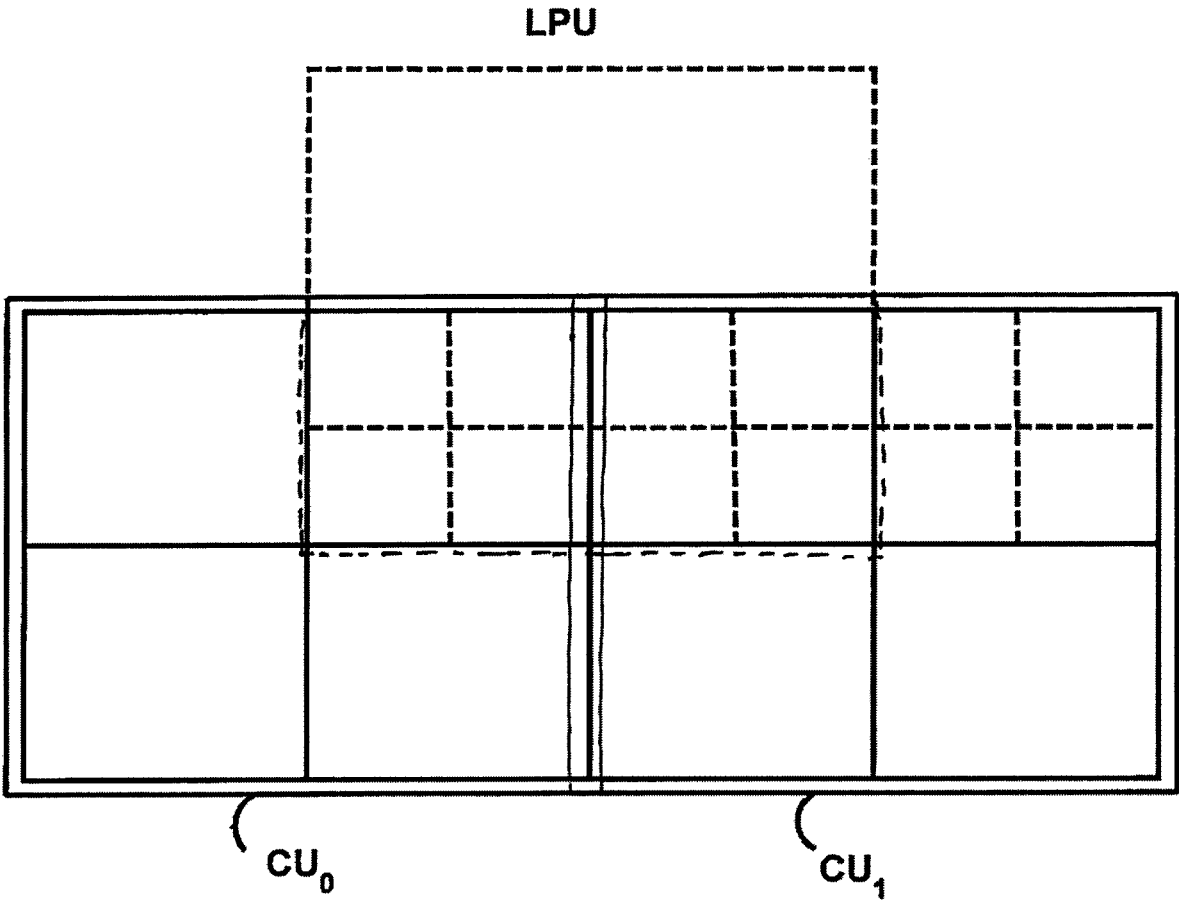


19/24

FIG. 14A

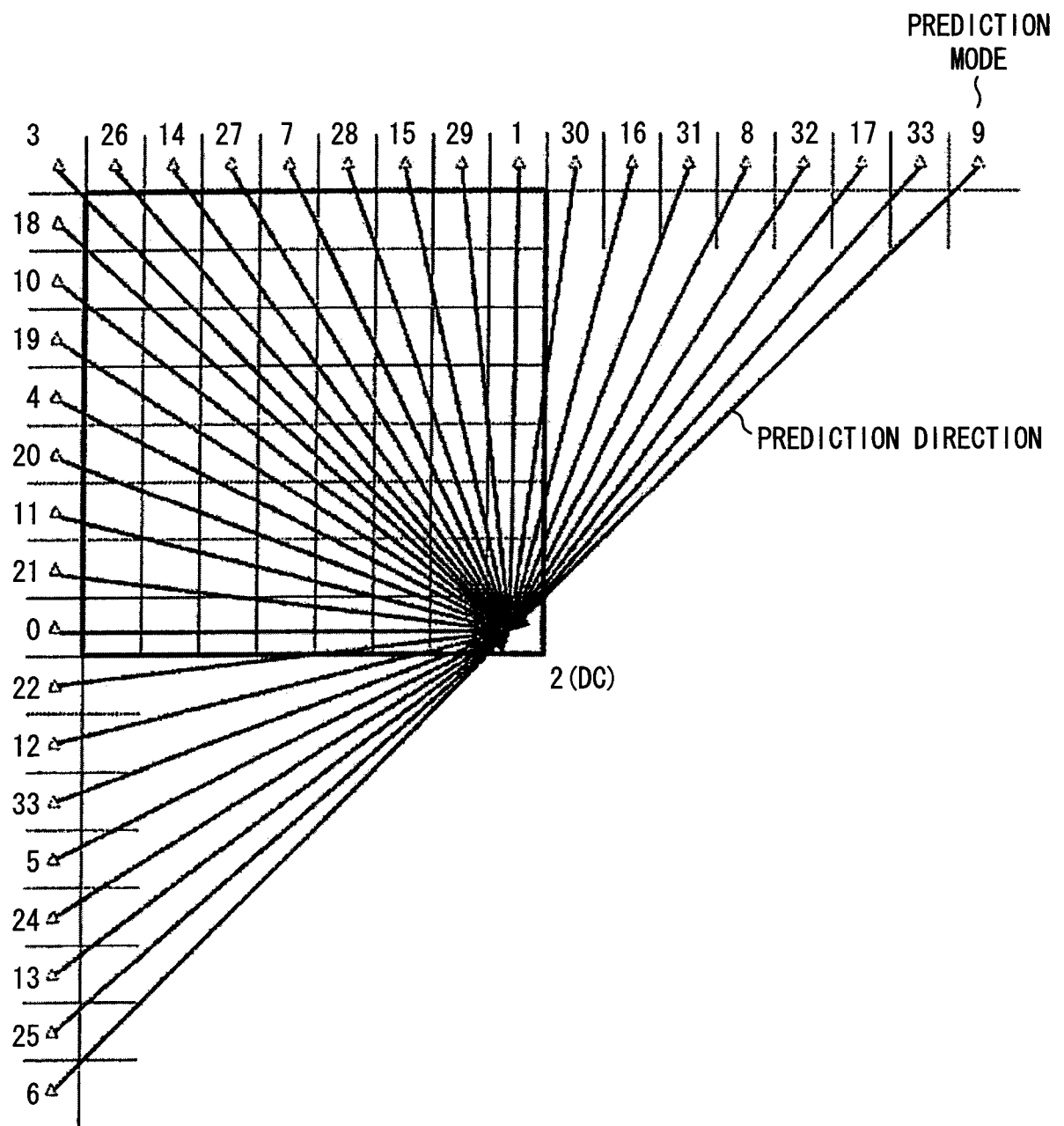


F I G. 1 4 B



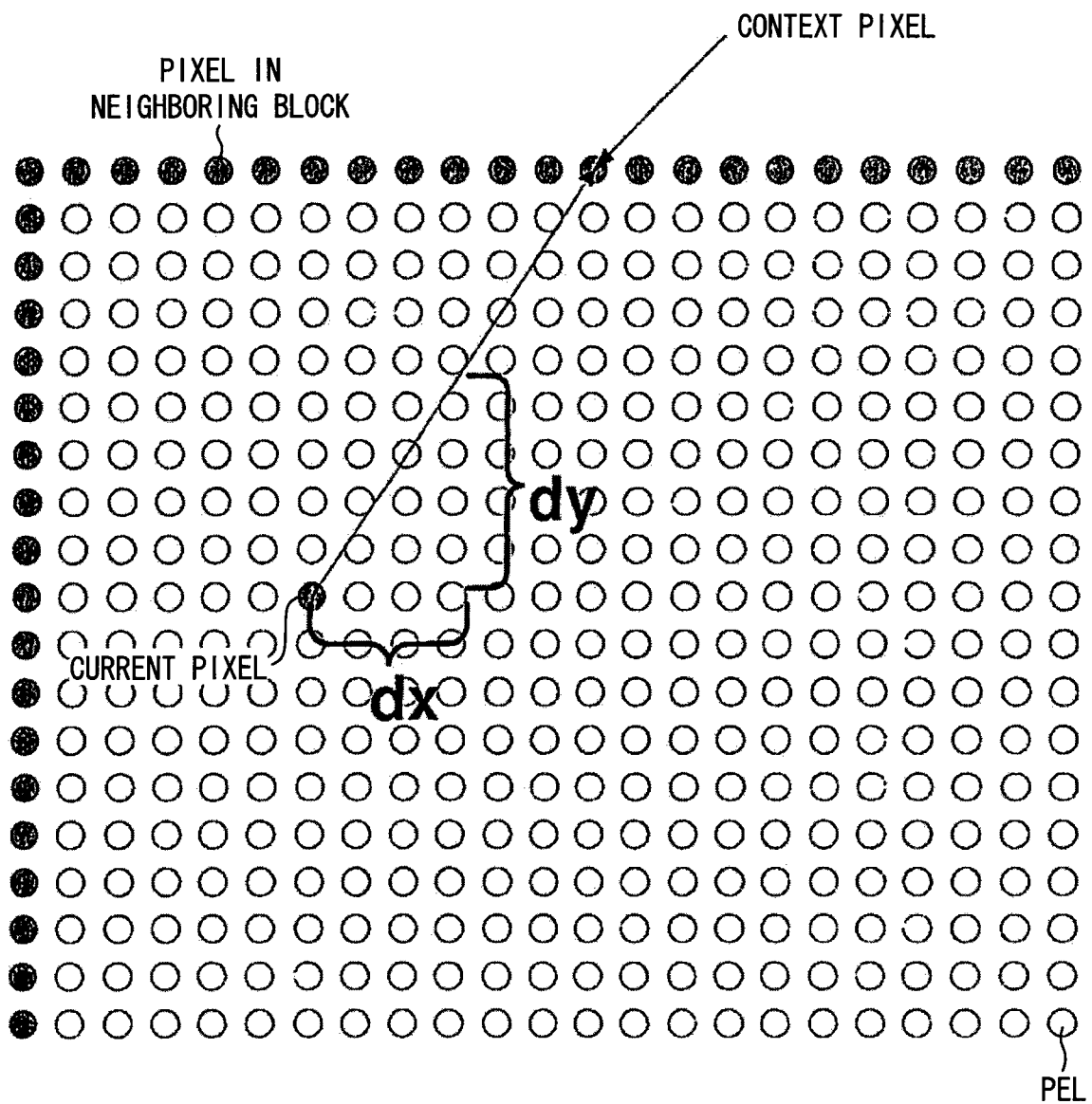
21/24

FIG. 15



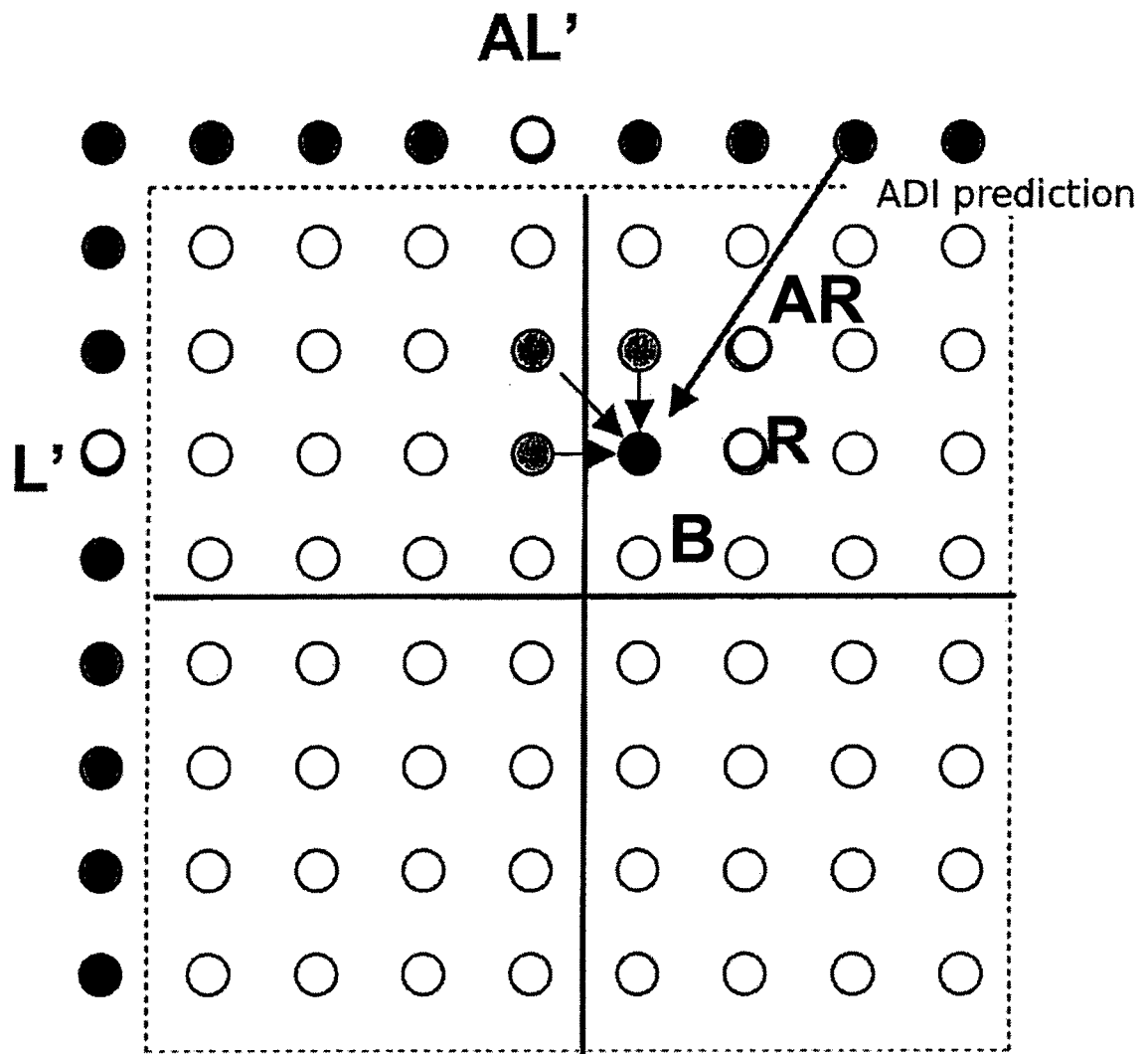
22/24

FIG. 16



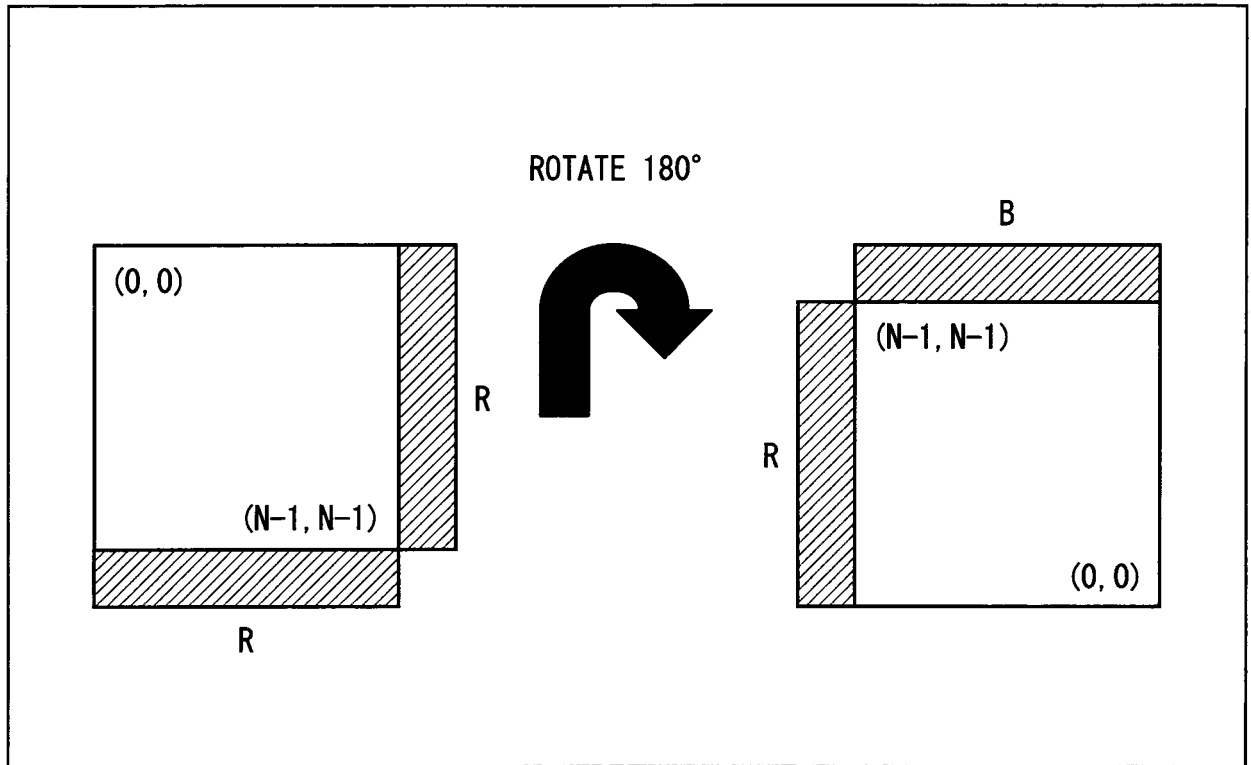
23/24

FIG. 17



24/24

FIG. 18



## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2011/066635

## 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. H04N7/24-7/68

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 1971-2011  
 Registered utility model specifications of Japan 1996-2011  
 Published registered utility model applications of Japan 1994-2011

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. |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Y         | Andrew Segall, Tomoyuki Yamamoto, Jie Zhao, Yoshihiro Kitaura, Yukinobu Yasugi, Tomohiro Ikai, "A Highly Efficient and Highly Parallel System for Video Coding", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting: Dresden, DE, 15-23 April, 2010, p.12-23, [JCTVC-A105r1] | 1-13                  |
| Y         | JP 2009-513056 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE)<br>2009.03.26, Fig.9<br>& US 2008/0285644 A1<br>& WO 2007/046644 A1<br>& KR 10-2007-0043638 A                                                                                                                                                               | 1-13                  |

☐ 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

"&amp;" document member of the same patent family

Date of the actual completion of the international search

18.10.2011

Date of mailing of the international search report

01.11.2011

Name and mailing address of the ISA/JP

**Japan Patent Office**

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

Authorized officer

**Daigoro BANDO**

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

5C 3241