Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2829158 C 2017/08/29

(11)(21) 2 829 158

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2012/03/07
(87) Date publication PCT/PCT Publication Date: 2012/09/13
(45) Date de délivrance/Issue Date: 2017/08/29
(85) Entrée phase nationale/National Entry: 2013/09/04
(86) N° demande PCT/PCT Application No.: US 2012/028093
(87) N° publication PCT/PCT Publication No.: 2012/122284
(30) Priorités/Priorities: 2011/03/08 (US61/450,555); 2011/03/10 (US61/451,496); 2011/03/10 (US61/451,485); 2011/03/14 (US61/452,384); 2011/06/08 (US61/494,855); 2011/06/15 (US61/497,345); 2012/03/06 (US13/413,526)

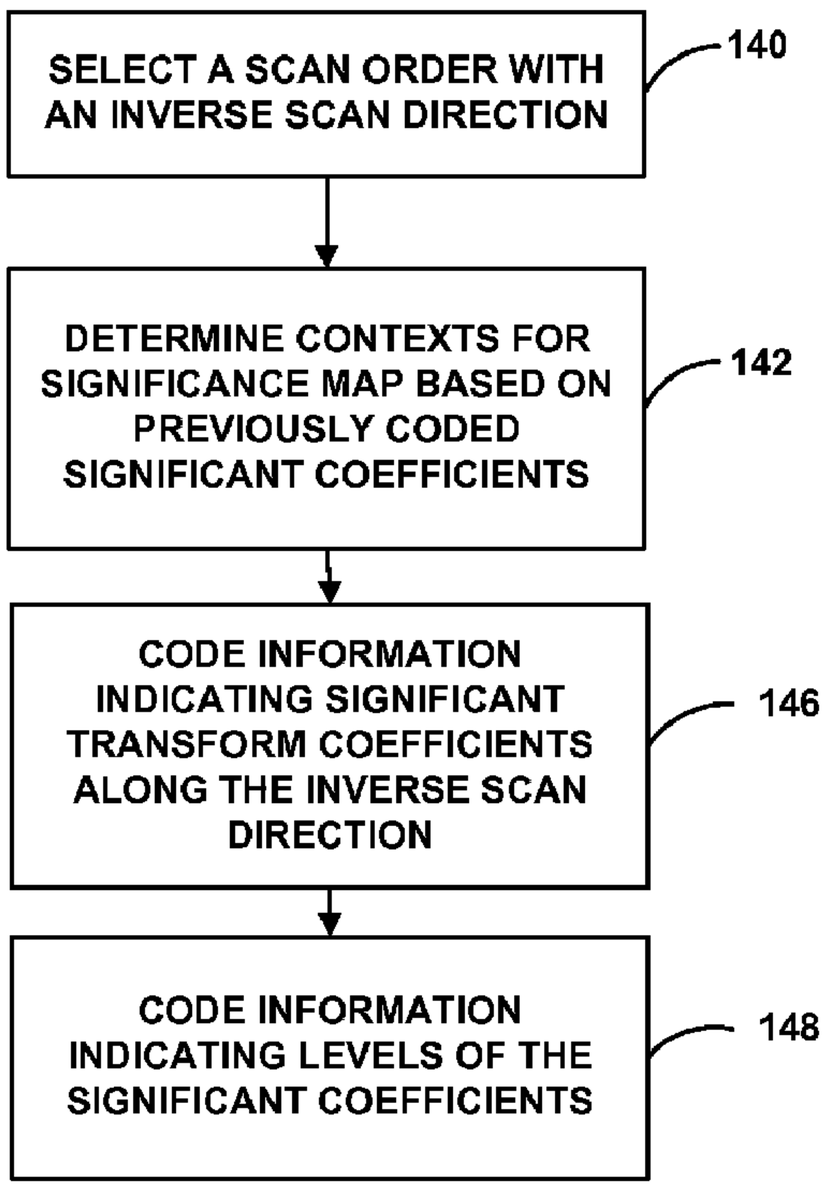
(51) Cl.Int./Int.Cl. *H04N 19/18* (2014.01), *H04N 19/129* (2014.01), *H04N 19/13* (2014.01), *H04N 19/176* (2014.01), *H04N 19/61* (2014.01)

(72) Inventeurs/Inventors:
SOLE ROJALS, JOEL, US;
JOSHI, RAJAN LAXMAN, US;
KARCZEWICZ, MARTA, US

(73) Propriétaire/Owner:
QUALCOMM INCORPORATED, US

(74) Agent: SMART & BIGGAR

(54) Titre : CODAGE DE COEFFICIENTS DE TRANSFORMEE POUR UN CODAGE VIDEO
(54) Title: CODING OF TRANSFORM COEFFICIENTS FOR VIDEO CODING



(57) Abrégé/Abstract:
This disclosure describes techniques for coding transform coefficients associated with a block of residual video data in a video coding process. Aspects of this disclosure include the selection of a scan order for both significance map coding and level coding,

CA 2829158 C 2017/08/29

(11)(21) 2 829 158

(13) C

(57) Abrégé(suite)/Abstract(continued):
as well as the selection of contexts for entropy coding consistent with the selected scan order. This disclosure proposes a harmonization of the scan order to code both the significance map of the transform coefficients as well as to code the levels of the transform coefficient. It is proposed that the scan order for the significance map should be in the inverse direction (i.e., from the higher frequencies to the lower frequencies). This disclosure also proposes that transform coefficients be scanned in subsets as opposed to fixed subblocks. In particular, transform coefficients are scanned in a subset consisting of a number of consecutive coefficients according to the scan order.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2012 (13.09.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/122284 A1

(51) International Patent Classification:
H04N 7/26 (2006.01) *H04N 7/50* (2006.01)
H04N 7/30 (2006.01)

(21) International Application Number:
PCT/US2012/028093

(22) International Filing Date:
7 March 2012 (07.03.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

61/450,555	8 March 2011 (08.03.2011)	US
61/451,485	10 March 2011 (10.03.2011)	US
61/451,496	10 March 2011 (10.03.2011)	US
61/452,384	14 March 2011 (14.03.2011)	US
61/494,855	8 June 2011 (08.06.2011)	US
61/497,345	15 June 2011 (15.06.2011)	US
13/413,526	6 March 2012 (06.03.2012)	US

(71) Applicant *(for all designated States except US)*: **QUALCOMM INCORPORATED** [US/US]; ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, Ca 92121-1714 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **SOLE ROJALS, Joel** [ES/US]; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US). **JOSHI, Rajan L.** [US/US]; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US). **KARCZEWICZ, Marta** [US/US]; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(74) Agent: **DAWLEY, Brian R.**; Shumaker & Sieffert, P.A., 1625 Radio Drive, Suite 300, Woodbury, MN 55125 (US).

(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, QA, RO, RS, RU, RW, SC, SD,

[Continued on next page]

(54) Title: CODING OF TRANSFORM COEFFICIENTS FOR VIDEO CODING

(57) Abstract: This disclosure describes techniques for coding transform coefficients associated with a block of residual video data in a video coding process. Aspects of this disclosure include the selection of a scan order for both significance map coding and level coding, as well as the selection of contexts for entropy coding consistent with the selected scan order. This disclosure proposes a harmonization of the scan order to code both the significance map of the transform coefficients as well as to code the levels of the transform coefficient. It is proposed that the scan order for the significance map should be in the inverse direction (i.e., from the higher frequencies to the lower frequencies). This disclosure also proposes that transform coefficients be scanned in subsets as opposed to fixed subblocks. In particular, transform coefficients are scanned in a subset consisting of a number of consecutive coefficients according to the scan order.

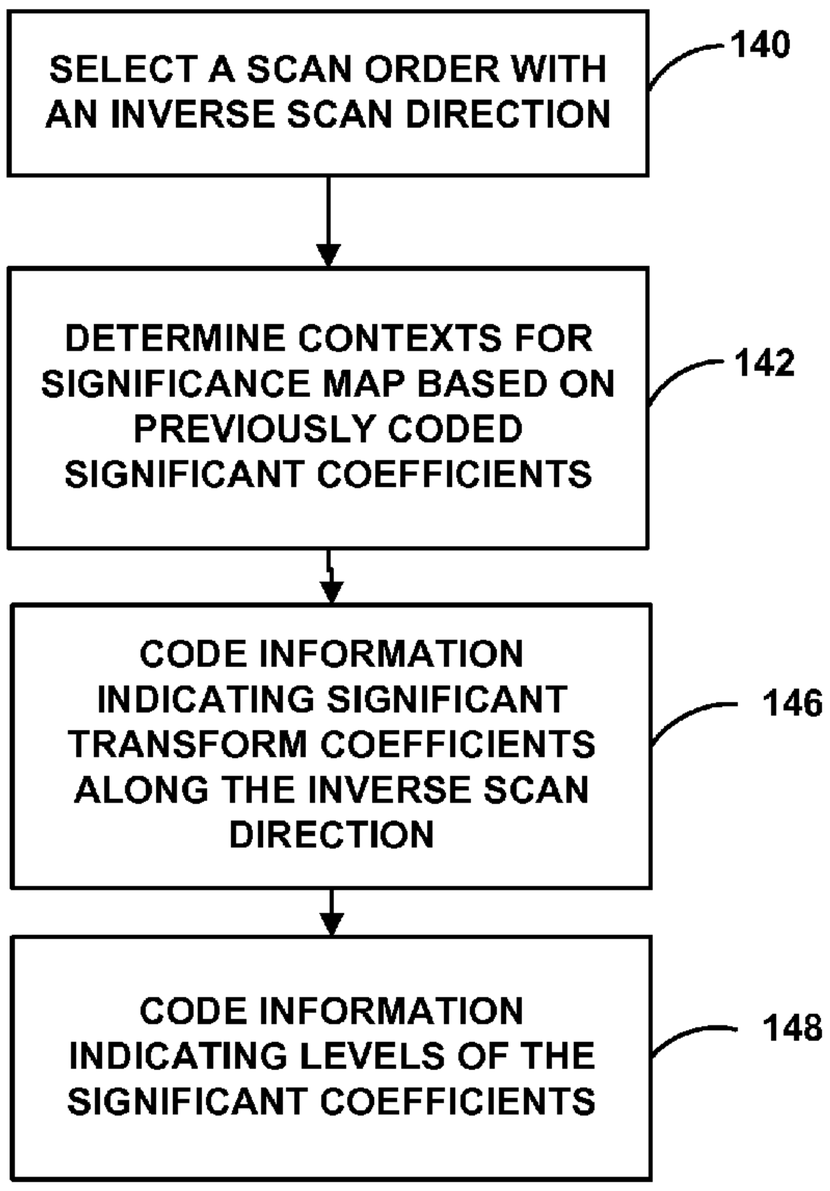


FIG. 26

WO 2012/122284 A1

WO 2012/122284 A1

SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

55158-20

CODING OF TRANSFORM COEFFICIENTS FOR VIDEO CODING

[0001]

TECHNICAL FIELD

[0002] This disclosure relates to video coding, and more particularly to techniques for scanning and coding transform coefficients generated by video coding processes.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, video teleconferencing devices, and the like. Digital video devices implement video compression techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), the High Efficiency Video Coding (HEVC) standard presently under development, and extensions of such standards, to transmit, receive and store digital video information more efficiently.

[0004] Video compression techniques include spatial prediction and/or temporal prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video frame or slice may be partitioned into blocks. Each block can be further partitioned. Blocks in an intra-coded (I) frame or slice are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same frame or slice. Blocks in an inter-coded (P or B) frame or slice may use spatial prediction with respect to reference samples in neighboring blocks in the same frame or slice or temporal prediction with respect to reference samples in other reference frames. Spatial or temporal prediction results in a predictive block for a block to be coded.

Residual data represents pixel differences between the original block to be coded and the predictive block.

[0005] An inter-coded block is encoded according to a motion vector that points to a block of reference samples forming the predictive block, and the residual data indicating the difference between the coded block and the predictive block. An intra-coded block is encoded according to an intra-coding mode and the residual data. For further compression, the residual data may be transformed from the pixel domain to a transform domain, resulting in residual transform coefficients, which then may be quantized. The quantized transform coefficients, initially arranged in a two-dimensional array, may be scanned in a particular order to produce a one-dimensional vector of transform coefficients for entropy coding.

SUMMARY

[0006] In general, this disclosure describes devices and methods for coding transform coefficients associated with a block of residual video data in a video coding process. The techniques, structures and methods described in this disclosure are applicable for video coding processes that use entropy coding (e.g., context adaptive binary arithmetic coding (CABAC)) to code the transform coefficients. Aspects of this disclosure include the selection of a scan order for both significance map coding and level and sign coding, as well as the selection of contexts for entropy coding consistent with the selected scan order. The techniques, structures and methods of this disclosure are applicable for use in both a video encoder and a video decoder.

[0007] This disclosure proposes a harmonization of the scan order to code both the significance map of the transform coefficients as well as to code the levels of the transform coefficient. That is to say, in some examples, the scan order for the significance map and the level coding should have the same pattern and direction. In another example, it is proposed that the scan order for the significance map should be in the inverse direction (i.e., from the coefficients for higher frequencies to the coefficients for lower frequencies). In yet another example, it is proposed that the scan order for the significance map and the level coding should be harmonized such that each proceeds in an inverse direction.

[0008] This disclosure also proposes that, in some examples, the transform coefficients be scanned in sub-sets. In particular, transform coefficients are scanned in a sub-set

consisting of a number of consecutive coefficients according to the scan order. Such sub-sets may be applicable for both the significance map scan as well as the coefficient level scan.

[0009] Additionally, this disclosure proposes that, in some examples, the significance map and the coefficient level scan are performed in consecutive scans and according to the same scan order. In one aspect, the scan order is an inverse scan order. The consecutive scans may consist of several scan passes. Each scan pass may consist of a syntax element scan pass. For example, a first scan is the significance map scan (also called bin 0 of the level of transform coefficients), a second scan is of bin one of the levels of transform coefficients in each sub-set, a third scan may be of bin two of the levels of transform coefficients in each sub-set, a fourth scan is of the remaining bins of the levels of transform coefficients, and a fifth scan is of the sign of the levels of transform coefficients. The sign pass may be at any point after the significance map pass. Additionally, the number of scan passes can be reduced by coding more than one syntax element per pass. For example, one scan pass for the syntax elements using coded bins and a second scan pass for the syntax elements using bypass bins (e.g., the remaining levels and the sign). In this context, a bin is part of the bin string that is entropy coded. A given a non-binary valued syntax element is mapped to a binary sequence (the so-called bin string).

[0010] This disclosure also proposes that, in some examples, the transform coefficients are entropy coded using CABAC in two different context regions. The context derivation for a first context region depends on the position of the transform coefficients while the context derivation for the second region depends on causal neighbors of the transform coefficients. In another example, the second context region can use two different context models depending on the location of the transform coefficients.

[0011] In one example of the disclosure, a method of coding transform coefficients associated with residual video data in a video coding process is proposed. The method comprises coding information indicating the significant transform coefficients in a block of transform coefficients with a scan proceeding in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients.

[0012] In another example of the disclosure, a system for coding transform coefficients associated with residual video data in a video coding process is proposed. The system comprises a video coding unit configured to code information indicating the significant

transform coefficients in a block of transform coefficients with a scan proceeding in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients.

[0013] In another example of the disclosure, a system for coding transform coefficients associated with residual video data in a video coding process is proposed. The system comprises means for coding information indicating the significant transform coefficients in a block of transform coefficients with a scan proceeding in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients, and means for forming a significance map for the block based on the coded information indicating the significant transform coefficients.

[0014] In another example of the disclosure, a computer program product comprises a computer-readable storage medium having stored thereon instructions that, when executed, cause a processor of a device for coding transform coefficients associated with residual video data in a video coding process to code information indicating the significant transform coefficients in a block of transform coefficients with a scan proceeding in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients.

[0014a] According to one aspect of the present invention, there is provided a method of coding transform coefficients associated with residual video data in a video coding process, the method comprising: coding a respective syntax element for each transform coefficient in a block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and coding information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

[0014b] According to another aspect of the present invention, there is provided a system for coding transform coefficients associated with residual video data in a video coding process, the system comprising: a memory configured to store a block of transform coefficients; and a video coding processor comprising an integrated circuit, the video coding processor configured to: code a respective syntax element for each transform coefficient in the block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and code information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

[0014c] According to still another aspect of the present invention, there is provided a system for coding transform coefficients associated with residual video data in a video coding process, the system comprising: means for coding a respective syntax element for each transform coefficient in a block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and means for coding information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

[0014d] According to yet another aspect of the present invention, there is provided a non-transitory computer-readable storage medium having stored thereon instructions that, when executed, cause one or more processors of a device for coding transform coefficients associated with residual video data in a video coding process to: code a respective syntax element for each transform coefficient in a block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding

transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and coding information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

[0015] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0016] FIG. 1 is a conceptual diagram illustrating a significance map coding process.

[0017] FIG. 2 is a conceptual diagram illustrating scanning patterns and directions for significance map coding.

[0018] FIG. 3 is a conceptual diagram illustrating a scanning technique for level coding of a transform unit.

[0019] FIG. 4 is a block diagram illustrating an example video coding system.

[0020] FIG. 5 is a block diagram illustrating an example video encoder.

[0021] FIG. 6 is a conceptual diagram illustrating inverse scan orders for significance map and coefficient level coding.

[0022] FIG. 7 is a conceptual diagram illustrating a first sub-set of transform coefficients according to an inverse diagonal scan order.

[0023] FIG. 8 is a conceptual diagram illustrating a first sub-set of transform coefficients according to an inverse horizontal scan order.

[0024] FIG. 9 is a conceptual diagram illustrating a first sub-set of transform coefficients according to an inverse vertical scan order.

[0025] FIG. 10 is a conceptual diagram illustrating context regions for significance map coding.

[0026] FIG. 11 is a conceptual diagram illustrating example context regions for significance map coding using an inverse scan order.

[0027] FIG. 12 is a conceptual diagram illustrating example causal neighbors for entropy coding using a forward scan order.

[0028] FIG. 13 is a conceptual diagram illustrating example causal neighbors for entropy coding using an inverse scan order.

[0029] FIG. 14 is a conceptual diagram illustrating example context regions for entropy coding using an inverse scan order.

[0030] FIG. 15 is a conceptual diagram illustrating example causal neighbors for entropy coding using an inverse scan order.

[0031] FIG. 16 is a conceptual diagram illustrating another example of context regions for CABAC using an inverse scan order.

[0032] FIG. 17 is a conceptual diagram illustrating another example of context regions for CABAC using an inverse scan order.

[0033] FIG. 18 is a conceptual diagram illustrating another example of context regions for CABAC using an inverse scan order.

[0034] FIG. 19 is a block diagram illustrating an example entropy coding unit.

[0035] FIG. 20 is a block diagram illustrating an example video decoder.

[0036] FIG. 21 is a block diagram illustrating an example entropy decoding unit.

[0037] FIG. 22 is a flowchart illustrating an example process for significance map and coefficients level scanning with a harmonized scan order.

[0038] FIG. 23 is a flowchart illustrating an example process for significance map and coefficients level scanning and entropy coding context derivation.

[0039] FIG. 24 is a flowchart illustrating another example process for significance map and coefficients level scanning and entropy coding context derivation.

[0040] FIG. 25 is a flowchart illustrating another example process for significance map and coefficients level scanning and entropy coding context derivation.

[0041] FIG. 26 is a flowchart illustrating an example process for significance map coding using an inverse scan direction.

[0042] FIG. 27 is a flowchart illustrating an example process for significance map and coefficient level scanning according to sub-sets of transform coefficients.

[0043] FIG. 28 is a flowchart illustrating another example process for significance map and coefficient level scanning according to sub-sets of transform coefficients.

[0044] FIG. 29 is a flowchart illustrating another example process for significance map and coefficient level scanning according to sub-sets of transform coefficients.

[0045] FIG. 30 is a flowchart illustrating an example process for entropy coding using multiple regions.

DETAILED DESCRIPTION

[0046] Digital video devices implement video compression techniques to transmit and receive digital video information more efficiently. Video compression may apply spatial (intra-frame) prediction and/or temporal (inter-frame) prediction techniques to reduce or remove redundancy inherent in video sequences.

[0047] For video coding according to the high efficiency video coding (HEVC) standard currently under development by the Joint Cooperative Team for Video Coding (JCT-VC), as one example, a video frame may be partitioned into coding units. A coding unit generally refers to an image region that serves as a basic unit to which various coding tools are applied for video compression. A coding unit is typically square (though not necessarily), and may be considered to be similar to a so-called macroblock, e.g., under other video coding standards such as ITU-T H.264. Coding according to some of the presently proposed aspects of the developing HEVC standard will be described in this application for purposes of illustration. However, the techniques described in this disclosure may be useful for other video coding processes, such as those defined according to H.264 or other standard or proprietary video coding processes.

[0048] To achieve desirable coding efficiency, a coding unit (CU) may have variable sizes depending on video content. In addition, a coding unit may be split into smaller blocks for prediction or transform. In particular, each coding unit may be further

partitioned into prediction units (PUs) and transform units (TUs). Prediction units may be considered to be similar to so-called partitions under other video coding standards, such as the H.264 standard. A transform unit (TU) generally refers to a block of residual data to which a transform is applied to produce transform coefficients.

[0049] A coding unit usually has a luminance component, denoted as Y, and two chroma components, denoted as U and V. Depending on the video sampling format, the size of the U and V components, in terms of number of samples, may be the same as or different from the size of the Y component.

[0050] To code a block (e.g., a prediction unit of video data), a predictor for the block is first derived. The predictor, also referred to as a predictive block, can be derived either through intra (I) prediction (i.e., spatial prediction) or inter (P or B) prediction (i.e. temporal prediction). Hence, some prediction units may be intra-coded (I) using spatial prediction with respect to reference samples in neighboring reference blocks in the same frame (or slice), and other prediction units may be uni-directionally inter-coded (P) or bi-directionally inter-coded (B) with respect to blocks of reference samples in other previously-coded frames (or slices). In each case, the reference samples may be used to form a predictive block for a block to be coded.

[0051] Upon identification of a predictive block, the difference between the original video data block and its predictive block is determined. This difference may be referred to as the prediction residual data, and indicates the pixel differences between the pixel values in the block to the coded and the pixel values in the predictive block selected to represent the coded block. To achieve better compression, the prediction residual data may be transformed, e.g., using a discrete cosine transform (DCT), an integer transform, a Karhunen-Loeve (K-L) transform, or another transform.

[0052] The residual data in a transform block, such as a TU, may be arranged in a two-dimensional (2D) array of pixel difference values residing in the spatial, pixel domain. A transform converts the residual pixel values into a two-dimensional array of transform coefficients in a transform domain, such as a frequency domain. For further compression, the transform coefficients may be quantized prior to entropy coding. An entropy coder then applies entropy coding, such as Context Adaptive Variable Length Coding (CAVLC), Context Adaptive Binary Arithmetic Coding (CABAC), Probability Interval Partitioning Entropy Coding (PIPE), or the like, to the quantized transform coefficients.

[0053] To entropy code a block of quantized transform coefficients, a scanning process is usually performed so that the two-dimensional (2D) array of quantized transform coefficients in a block is processed, according to a particular scan order, in an ordered, one-dimensional (1D) array, i.e., vector, of transform coefficients. Entropy coding is applied in the 1-D order of transform coefficients. The scan of the quantized transform coefficients in a transform unit serializes the 2D array of transform coefficients for the entropy coder. A significance map may be generated to indicate the positions of significant (i.e., non-zero) coefficients. Scanning may be applied to scan levels of significant (i.e., nonzero) coefficients, and/or to code signs of the significant coefficients.

[0054] For a DCT, as an example, there is often a higher probability of non-zero coefficients toward an upper left corner (i.e., a low frequency region) of the 2D transform unit. It may be desirable to scan the coefficients in a way that increases the probability of grouping non-zero coefficients together at one end of the serialized run of coefficients, permitting zero-valued coefficients to be grouped together toward another end of the serialized vector and more efficiently coded as runs of zeros. For this reason, scan order may be important for efficient entropy coding.

[0055] As one example, the so-called diagonal (or wavefront) scan order has been adopted for use in scanning quantized transform coefficients in the HEVC standard. Alternatively, zig-zag, horizontal, vertical or other scan orders may be used. Through transform and quantization, as mentioned above, non-zero transform coefficients are generally located at the low frequency area toward the upper left region of the block for an example in which the transform is a DCT. As a result, after the diagonal scanning process, which may traverse the upper left region first, non-zero transform coefficients are usually more likely to be located in the front portion of the scan. For a diagonal scanning process that traverses from the lower right region first, the non-zero transform coefficients are usually more likely to be located in the back portion of the scan.

[0056] A number of zero coefficients will typically be grouped at one end of the scan, depending on the scan direction, due to reduced energy at higher frequencies, and due to the effects of quantization, which may cause some nonzero coefficients to become zero-valued coefficients upon reduction of bit depth. These characteristics of coefficient distribution in the serialized 1D array may be utilized in entropy coder design to improve coding efficiency. In other words, if non-zero coefficients can be effectively

arranged in one portion of the 1D array through some appropriate scan order, better coding efficiency can be expected due to the design of many entropy coders.

[0057] To achieve this objective of placing more non-zero coefficients at one end of the 1D array, different scan orders may be used in a video encoder-decoder (CODEC) to code transform coefficients. In some cases, diagonal scanning may be effective. In other cases, different types of scanning, such as zig-zag, vertical or horizontal scanning may be more effective.

[0058] Different scan orders may be produced in a variety of ways. One example is that, for each block of transform coefficients, a "best" scan order may be chosen from a number of available scan orders. A video encoder then may provide an indication to the decoder, for each block, of an index of the best scan order among a set of scan orders denoted by respective indices. The selection of the best scan order may be determined by applying several scan orders and selecting one that is most effective in placing nonzero coefficients near the beginning or end of the 1D vector, thereby promoting efficient entropy coding.

[0059] In another example, the scan order for a current block may be determined based on various factors relating to the coding of the pertinent prediction unit, such as the prediction mode (I, B, P), block size, transform or other factors. In some cases, because the same information, e.g., prediction mode, can be inferred at both the encoder and decoder side, there may be no need to provide an indication of the scan order index to the decoder. Instead, the video decoder may store configuration data that indicates the appropriate scan order given knowledge of the prediction mode for a block, and one or more criteria that maps a prediction mode to a particular scan order.

[0060] To further improve coding efficiency, the available scan orders may not be constant all of the time. Instead, some adaptation might be enabled so that the scan order is adaptively adjusted, e.g., based on coefficients that are already coded. In general, the scan order adaptation may be done in such a way that, according to the selected scan order, zero and non-zero coefficients are more likely to be grouped together.

[0061] In some video CODECs, the initial available scan orders may be in a very regular form such as purely horizontal, vertical, diagonal, or zig-zag scan. Alternatively, the scan orders may be derived through a training process and therefore may appear to be somewhat random. The training process may involve application of different scan orders to a block or series of blocks to identify a scan order that produces desirable

results, e.g., in terms of efficient placement of nonzero and zero-valued coefficients, as mentioned above.

[0062] If a scan order is derived from a training process, or if a variety of different scan orders can be selected, it may be beneficial to save the particular scan orders at both the encoder and decoder side. The amount of data specifying such scan orders can be substantial. For example, for a 32x32 transform block, one scan order may contain 1024 transform coefficient positions. Because there may be differently sized blocks and, for each size of transform block, there may be a number of different scan orders, the total amount of data that needs to be saved is not negligible. Regular scan orders such as diagonal, horizontal, vertical or zig-zag order may not require storage, or may require minimal storage. However, diagonal, horizontal, vertical or zig-zag orders may not provide sufficient variety to provide coding performance that is on par with trained scan orders.

[0063] In one conventional example, for H.264 and the HEVC standard presently under development, when the CABAC entropy coder is used, the positions of the significant coefficients (i.e., nonzero transform coefficients) in the transform block (i.e., transform unit in HEVC) are encoded prior to the levels of the coefficients. The process of coding the locations of the significant coefficients is called significance map coding. The significance of a coefficient is the same as the bin zero of the coefficient level. As shown in FIG. 1, significance map coding of the quantized transform coefficients 11 produces a significance map 13. The significance map 13 is a map of ones and zeros, where the ones indicate locations of significant coefficients. The significance map typically requires a high percentage of the video bit-rate. The techniques of this disclosure may also be applicable for use with other entropy coders (e.g., PIPE).

[0064] An example process for coding a significance map is described in D. Marpe, H. Schwarz, and T. Wiegand "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 7, July 2003. In this process, the significance map is coded if there is at least one significant coefficient in the block, as indicated by the Coded Block Flag (CBF), which is defined as:

> *Coded Block Flag: coded_block_flag is a one-bit symbol, which indicates if there are significant, i.e., nonzero coefficients inside a single block of transform coefficients, for which the coded block pattern*

indicates nonzero entries. If coded_block_flag is zero, no further information is transmitted for the related block.

[0065] If there are significant coefficients in the block, the significance map is encoded by following a scan order of transform coefficients in the block as follows:

Scanning of Transform Coefficients: two-dimensional arrays of transform coefficient levels of sub-blocks for which the coded_block_flag indicates nonzero entries are first mapped into a one-dimensional list using a given scanning pattern. In other words, sub-blocks with significant coefficients are scanned according to a scanning pattern.

[0066] Given the scanning pattern, the significance map is scanned as follows:

Significance Map: If the coded_block_flag indicates that a block has significant coefficients, a binary-valued significance map is encoded. For each transform coefficient in the scanning order, a one-bit symbol significant_coeff_flag is transmitted. If the significant_coeff_flag symbol is one, i.e., if a nonzero coefficient exists at this scanning position, a further one-bit symbol last_significant_coeff_flag is sent. This symbol indicates if the current significant coefficient is the last one inside the block or if further significant coefficients follow. If the last scanning position is reached and the significance map encoding was not already terminated by a last_significant_coeff_flag with value one, it is apparent that the last coefficient has to be significant.

[0067] Recent proposals for HEVC have removed the last_significant_coeff flag. In those proposals, before sending the significance map, an indication of the X and Y position of the position of the last significant coefficient is sent.

[0068] Currently, in HEVC, it is proposed that three scan patterns be used for the significance map: diagonal, vertical, and horizontal. FIG. 2 shows an example of a zig-zag scan 17, a vertical scan 19, a horizontal scan 21, and a diagonal scan 15. As shown in FIG. 2, each of these scans proceeds in a forward direction, i.e., from lower frequency transform coefficients in the upper left corner of the transform block to the higher

frequency transform coefficients in the lower right corner of the transform block. After the significance map is coded, the remaining level information (bins 1 – N, where N is the total number of bins) for each significant transform coefficient (i.e., the coefficient value) is coded.

[0069] In the CABAC process previously specified in the H.264 standard, following the handling of 4x4 sub-blocks, each of the transform coefficient levels is binarized, e.g., according to a unary code, to produce a series of bins. In H.264, the CABAC context model set for each sub-block consists of two times five context models with five models for both the first bin and all remaining bins (up to and including the 14^{th} bin) of the coeff_abs_level_minus_one syntax element, which encodes the absolute value of a transform coefficient. Notably, in one proposed version of HEVC, the remaining bins include only bin 1 and bin 2. The remainder of the coefficient levels are coded with Golomb-Rice coding and exponential Golomb codes.

[0070] In HEVC, the selection of context models may be performed as in the original CABAC process proposed for the H.264 standard. However, different sets of context models may be selected for different sub-blocks. In particular, the choice of the context model set for a given sub-block depends on certain statistics of the previously coded sub-blocks.

[0071] FIG. 3 shows the scanning order followed by one proposed version of the HEVC process to encode the levels of transform coefficients (absolute value of the level and sign of the level) in a transform unit 25. Note that there is a forward zig-zag pattern 27 for scanning of the 4x4 sub-blocks of a larger block, and an inverse zig-zag pattern 23 for scanning the levels of transform coefficients within each sub-block. In other words, a series of 4x4 sub-blocks are scanned in a forward zig-zag pattern such that the sub-blocks are scanned in a sequence. Then, within each sub-block, an inverse zig-zag scan is performed to scan the levels of the transform coefficients within the sub-block. Hence, the transform coefficients in the two-dimensional array formed by the transform unit are serialized into a one-dimensional array such that coefficients that are inverse scanned in a given sub-block are then followed by coefficients that are inverse scanned in a successive sub-block.

[0072] In one example, the CABAC coding of coefficients scanned according to the sub-block scan approach shown in FIG. 3 may use 60 contexts, i.e., 6 sets of 10 contexts each, distributed as described below. For a 4x4 block, 10 context models might be used (5 models for bin 1 and 5 models for bins 2 to 14), as shown in Table 1:

Model bin 1	
0	Encoded coefficient larger than 1 Initial - no trailing ones in sub-block
1	
2	1 trailing one in sub-block
3	2 trailing ones in sub-block
4	3 or more trailing ones in sub-block

Model bin 2-14 (remaining bins)	
0	Initial or 0 coefficients larger than one
1	1 coefficient larger than one
2	2 coefficients larger than one
3	3 coefficients larger than one
4	4 or more coefficient larger than one

Table 1-Contexts for bin 1 and bins 2 to 14 of the coefficient levels of a sub-block

[0073] Per Table 1, one of context models 0-4 in the context set is used for bin 1 if, respectively, the currently encoded coefficient that is being scanned in the sub-block is encoded after a coefficient larger than 1 has been encoded within the sub-block, the currently encoded coefficient is the initial coefficient scanned in the sub-block or there are no trailing ones (no previously encoded coefficients) in the sub-block, there is one trailing one in the sub-block (i.e., a one has been encoded but no coefficients larger than one have been encoded), there are two trailing ones in the sub-block, or there are three or more trailing ones in the sub-block. For each of bins 2-14 (although the currently proposed version of HEVC codes only bin 2 using CABAC, with successive bins of the coefficient level being coded with an exponential Golomb code), one of context models 0-4 may be used, respectively, if the coefficient is the initial coefficient scanned in the sub-block or there are zero previously coded coefficients larger than one, there is one previously coded coefficient larger than one, there are two previously coded coefficients larger than one, there are three previously coded coefficients larger than one, or there are four previously coded coefficients larger than one.

[0074] There are 6 different sets of these 10 models, depending on the number of coefficients larger than 1 in the previous coded 4x4 sub-block in the forward scan of sub-blocks:

Context Set	
0	For block size 4x4 only
1	0-3 Coefficients Larger than 1 in previous sub-block
2	4-7 LargerT1 in previous sub-block
3	8-11 LargerT1 in previous sub-block
4	12-15 LargerT1 in previous sub-block
5	**First 4x4 sub-block** 16 LargerT1 in previous sub-block

Table 2-Contexts for bin 1 and bins 2 to 14

[0075] Per Table 2, sets 0-5 of context models are used for a given sub-block if, respectively, the sub-block size is 4x4, there are 0 to 3 coefficients larger than 1 in the previously coded sub-block, there are 4 to 7 coefficients larger than 1 in the previously coded sub-block, there are 8-11 coefficients larger than 1 in the previously coded sub-block, there are 12 to 15 coefficients larger than 1 in the previously coded sub-block, or the given sub-block is the first 4x4 sub-block (top left sub-block) or there are 16 coefficients larger than 1 in the previously coded sub-block.

[0076] The above-described coding process for H.264 and that currently proposed for HEVC has several drawbacks. As shown in FIG. 3, one drawback is that the scan for coefficient levels proceeds forward for the scan of sub-blocks (i.e., starting with the upper left sub-block) but then backward for the scan of the coefficient levels within each sub-block (i.e., starting with the lower right coefficient in each sub-block). This approach implies going back and forth within the block, which may make data fetching more complex.

[0077] Another drawback comes from the fact that the scan order of the coefficient level is different from the scan order of the significance map. In HEVC, there are three different proposed scan orders for the significance map: forward diagonal, forward horizontal and forward vertical as shown in FIG. 2. All of the significant coefficient scans are different from the scan of coefficient levels currently proposed for HEVC, since the level scans proceed in an inverse direction. Because the direction and pattern of the coefficient level scan is not matched with the direction and pattern of the significance scan, more coefficient levels have to be checked. For instance, assume that a horizontal scan is used for the significance map, and the last significant coefficient is found at the end of the first row of coefficients. The coefficient level scan in HEVC would require a diagonal scan across multiple rows for the level scan, when only the first row actually contains coefficient levels different from 0. Such a scanning process may introduce unwanted inefficiencies.

[0078] In the current proposal for HEVC, the scan of the significance map proceeds forward in the block, from the DC coefficient found in the upper left corner of the block to the highest frequency coefficient typically found in the lower right corner of the block, while the scan for the coefficient levels is backward within each 4x4 sub-block. This too may result in more complex and more inefficient data fetching.

[0079] Another drawback to current HEVC proposals comes from the context sets. The context set (see Tables 2 above) for CABAC is different for block size 4x4 than for other block sizes. In accordance with this disclosure, it would be desirable to harmonize contexts across all block sizes so that less memory is dedicated to storing different context sets.

[0080] Also, as will be described in more detail below, the currently proposed CABAC contexts for the significance map for HEVC are only valid if the scan order is forward. As such, this would not allow for inverse significance map scans.

[0081] Furthermore, the contexts described above for encoding the level of a quantized coefficient attempt to exploit the local correlation of coefficient levels. These contexts depend on the correlation among 4x4 sub-blocks (see the contexts set in Table 2), and the correlation within each sub-block (see context models in Table 1). The drawback of these contexts is that the dependency might be too far (i.e., there is low dependency between coefficients which are separated from one another by several other coefficients, from one sub-block to another). Also, within each sub-block, the dependency might be weak.

[0082] This disclosure proposes several different features that may reduce or eliminate some of the drawbacks described above. In some examples, these features may provide a more efficient and harmonized scan order of the transform coefficients in video coding. In other examples of this disclosure, these features provide a more efficient set of contexts to be used in CABAC-based entropy coding of the transform coefficients consistent with the proposed scan order. It should be noted that all of the techniques described in this disclosure may be used independently or may be used together in any combination.

[0083] FIG. 4 is a block diagram illustrating an example video encoding and decoding system 10 that may be configured to utilize techniques for coding transform coefficients in accordance with examples of this disclosure. As shown in FIG. 4, the system 10 includes a source device 12 that transmits encoded video to a destination device 14 via a communication channel 16. Encoded video may also be stored on a storage medium 34 or a file server 36 and may be accessed by the destination device 14 as desired. The source device 12 and the destination device 14 may comprise any of a wide variety of devices, including desktop computers, notebook (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called smartphones, televisions, cameras, display devices, digital media players, video gaming consoles, or the like. In many cases, such devices may be equipped for wireless communication. Hence, the communication channel 16 may comprise a wireless channel, a wired channel, or a combination of wireless and wired channels suitable for transmission of encoded video data. Similarly, the file server 36 may be accessed by the destination device 14 through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., DSL, cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server.

[0084] Techniques for coding transform coefficients, in accordance with examples of this disclosure, may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, streaming video transmissions, e.g., via the Internet, encoding of digital video for storage on a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, the system 10 may be configured to support one-way or two-way video transmission to

support applications such as video streaming, video playback, video broadcasting, and/or video telephony.

[0085] In the example of FIG. 4, the source device 12 includes a video source 18, a video encoder 20, a modulator/demodulator 22 and a transmitter 24. In the source device 12, the video source 18 may include a source such as a video capture device, such as a video camera, a video archive containing previously captured video, a video feed interface to receive video from a video content provider, and/or a computer graphics system for generating computer graphics data as the source video, or a combination of such sources. As one example, if the video source 18 is a video camera, the source device 12 and the destination device 14 may form so-called camera phones or video phones. However, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications.

[0086] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video information may be modulated by the modem 22 according to a communication standard, such as a wireless communication protocol, and transmitted to the destination device 14 via the transmitter 24. The modem 22 may include various mixers, filters, amplifiers or other components designed for signal modulation. The transmitter 24 may include circuits designed for transmitting data, including amplifiers, filters, and one or more antennas.

[0087] The captured, pre-captured, or computer-generated video that is encoded by the video encoder 20 may also be stored onto a storage medium 34 or a file server 36 for later consumption. The storage medium 34 may include Blu-ray discs, DVDs, CD-ROMs, flash memory, or any other suitable digital storage media for storing encoded video. The encoded video stored on the storage medium 34 may then be accessed by the destination device 14 for decoding and playback.

[0088] The file server 36 may be any type of server capable of storing encoded video and transmitting that encoded video to the destination device 14. Example file servers include a web server (e.g., for a website), an FTP server, network attached storage (NAS) devices, a local disk drive, or any other type of device capable of storing encoded video data and transmitting it to a destination device. The transmission of encoded video data from the file server 36 may be a streaming transmission, a download transmission, or a combination of both. The file server 36 may be accessed by the destination device 14 through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired

connection (e.g., DSL, cable modem, Ethernet, USB, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server.

[0089] The destination device 14, in the example of FIG. 4, includes a receiver 26, a modem 28, a video decoder 30, and a display device 32. The receiver 26 of the destination device 14 receives information over the channel 16, and the modem 28 demodulates the information to produce a demodulated bitstream for the video decoder 30. The information communicated over the channel 16 may include a variety of syntax information generated by the video encoder 20 for use by the video decoder 30 in decoding video data. Such syntax may also be included with the encoded video data stored on the storage medium 34 or the file server 36. Each of the video encoder 20 and the video decoder 30 may form part of a respective encoder-decoder (CODEC) that is capable of encoding or decoding video data.

[0090] The display device 32 may be integrated with, or external to, the destination device 14. In some examples, the destination device 14 may include an integrated display device and also be configured to interface with an external display device. In other examples, the destination device 14 may be a display device. In general, the display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0091] In the example of FIG. 4, the communication channel 16 may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines, or any combination of wireless and wired media. The communication channel 16 may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication channel 16 generally represents any suitable communication medium, or collection of different communication media, for transmitting video data from the source device 12 to the destination device 14, including any suitable combination of wired or wireless media. The communication channel 16 may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from the source device 12 to the destination device 14.

[0092] The video encoder 20 and the video decoder 30 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard presently under development, and may conform to the HEVC Test Model (HM). Alternatively, the video encoder 20 and the video decoder 30 may operate

according to other proprietary or industry standards, such as the ITU-T H.264 standard, alternatively referred to as MPEG-4, Part 10, Advanced Video Coding (AVC), or extensions of such standards. The techniques of this disclosure, however, are not limited to any particular coding standard. Other examples include MPEG-2 and ITU-T H.263.

[0093] Although not shown in FIG. 4, in some aspects, the video encoder 20 and the video decoder 30 may each be integrated with an audio encoder and decoder, and may include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, in some examples, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0094] The video encoder 20 and the video decoder 30 each may be implemented as any of a variety of suitable encoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of the video encoder 20 and the video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device.

[0095] The video encoder 20 may implement any or all of the techniques of this disclosure to improve encoding of transform coefficients in a video coding process. Likewise, the video decoder 30 may implement any or all of these techniques to improve decoding of transform coefficients in a video coding process. A video coder, as described in this disclosure, may refer to a video encoder or a video decoder. Similarly, a video coding unit may refer to a video encoder or a video decoder. Likewise, video coding may refer to video encoding or video decoding.

[0096] In one example of the disclosure, a video coder (such as video encoder 20 or video decoder 30) may be configured to code a plurality of transform coefficients associated with residual video data in a video coding process. The video coder may be configured to code information indicating significant coefficients for the plurality of transform coefficients according to a scan order, and code information indicating levels of the plurality transform coefficients according to the scan order.

[0097] In another example of the disclosure, a video coder (such as video encoder 20 or video decoder 30) may be configured to code a plurality of transform coefficients associated with residual video data in a video coding process. The video coder may be configured to code information indicating the significant transform coefficients in a block of transform coefficients with a scan proceeding in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients.

[0098] In another example of the disclosure, a video coder (such as video encoder 20 or video decoder 30) may be configured to code a plurality of transform coefficients associated with residual video data in a video coding process. The video coder may be configured to arrange a block of transform coefficients into one or more sub-sets of transform coefficients based on a scan order, code a first portion of levels of transform coefficients in each sub-set, wherein the first portion of levels includes at least a significance of the transform coefficients in each subset, and code a second portion of levels of transform coefficients in each sub-set.

[0099] In another example of the disclosure, a video coder (such as video encoder 20 or video decoder 30) may be configured to code information indicating significant coefficients for the plurality of transform coefficients according to a scan order, divide the coded information into at least a first region and a second region, entropy code the coded information in the first region according to a first set of contexts using context derivation criteria, and entropy code the coded information in the second region according to a second set of contexts using the same context derivation criteria as the first region.

[0100] FIG. 5 is a block diagram illustrating an example of a video encoder 20 that may use techniques for coding transform coefficients as described in this disclosure. The video encoder 20 will be described in the context of HEVC coding for purposes of illustration, but without limitation of this disclosure as to other coding standards or methods that may require scanning of transform coefficients. The video encoder 20 may perform intra- and inter-coding of CUs within video frames. Intra-coding relies on spatial prediction to reduce or remove spatial redundancy in video within a given video frame. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy between a current frame and previously coded frames of a video sequence. Intra-mode (I-mode) may refer to any of several spatial-based video compression modes. Inter-modes such as uni-directional prediction (P-mode) or bi-directional

prediction (B-mode) may refer to any of several temporal-based video compression modes.

[0101] As shown in FIG. 5, the video encoder 20 receives a current video block within a video frame to be encoded. In the example of FIG. 5, the video encoder 20 includes a motion compensation unit 44, a motion estimation unit 42, an intra-prediction module 46, a reference frame buffer 64, a summer 50, a transform module 52, a quantization unit 54, and an entropy encoding unit 56. The transform module 52 illustrated in FIG. 5 is the module that applies the actual transform to a block of residual data, and is not to be confused with block of transform coefficients, which also may be referred to as a transform unit (TU) of a CU. For video block reconstruction, the video encoder 20 also includes an inverse quantization unit 58, an inverse transform module 60, and a summer 62. A deblocking filter (not shown in FIG. 5) may also be included to filter block boundaries to remove blockiness artifacts from reconstructed video. If desired, the deblocking filter would typically filter the output of the summer 62.

[0102] During the encoding process, the video encoder 20 receives a video frame or slice to be coded. The frame or slice may be divided into multiple video blocks, e.g., largest coding units (LCUs). The motion estimation unit 42 and the motion compensation unit 44 perform inter-predictive coding of the received video block relative to one or more blocks in one or more reference frames to provide temporal compression. The intra-prediction module 46 may perform intra-predictive coding of the received video block relative to one or more neighboring blocks in the same frame or slice as the block to be coded to provide spatial compression.

[0103] The mode select unit 40 may select one of the coding modes, intra or inter, e.g., based on error (i.e., distortion) results for each mode, and provides the resulting intra- or inter-coded block to the summer 50 to generate residual block data and to the summer 62 to reconstruct the encoded block for use in a reference frame. Some video frames may be designated as I-frames, where all blocks in an I-frame are encoded in an intra-prediction mode. In some cases, the intra-prediction module 46 may perform intra-prediction encoding of a block in a P- or B-frame, e.g., when motion search performed by the motion estimation unit 42 does not result in a sufficient prediction of the block.

[0104] The motion estimation unit 42 and the motion compensation unit 44 may be highly integrated, but are illustrated separately for conceptual purposes. Motion estimation is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a prediction

unit in a current frame relative to a reference sample of a reference frame. A reference sample may be a block that is found to closely match the portion of the CU including the PU being coded in terms of pixel difference, which may be determined by sum of absolute difference (SAD), sum of square difference (SSD), or other difference metrics. Motion compensation, performed by the motion compensation unit 44, may involve fetching or generating values for the prediction unit based on the motion vector determined by motion estimation. Again, the motion estimation unit 42 and the motion compensation unit 44 may be functionally integrated, in some examples.

[0105] The motion estimation unit 42 calculates a motion vector for a prediction unit of an inter-coded frame by comparing the prediction unit to reference samples of a reference frame stored in the reference frame buffer 64. In some examples, the video encoder 20 may calculate values for sub-integer pixel positions of reference frames stored in the reference frame buffer 64. For example, the video encoder 20 may calculate values of one-quarter pixel positions, one-eighth pixel positions, or other fractional pixel positions of the reference frame. Therefore, the motion estimation unit 42 may perform a motion search relative to the full pixel positions and fractional pixel positions and output a motion vector with fractional pixel precision. The motion estimation unit 42 sends the calculated motion vector to the entropy encoding unit 56 and the motion compensation unit 44. The portion of the reference frame identified by a motion vector may be referred to as a reference sample. The motion compensation unit 44 may calculate a prediction value for a prediction unit of a current CU, e.g., by retrieving the reference sample identified by a motion vector for the PU.

[0106] The intra-prediction module 46 may intra-prediction encode the received block, as an alternative to inter-prediction performed by the motion estimation unit 42 and the motion compensation unit 44. The intra-prediction module 46 may encode the received block relative to neighboring, previously coded blocks, e.g., blocks above, above and to the right, above and to the left, or to the left of the current block, assuming a left-to-right, top-to-bottom encoding order for blocks. The intra-prediction module 46 may be configured with a variety of different intra-prediction modes. For example, the intra-prediction module 46 may be configured with a certain number of directional prediction modes, e.g., 33 directional prediction modes, based on the size of the CU being encoded.

[0107] The intra-prediction module 46 may select an intra-prediction mode by, for example, calculating error values for various intra-prediction modes and selecting a

mode that yields the lowest error value. Directional prediction modes may include functions for combining values of spatially neighboring pixels and applying the combined values to one or more pixel positions in a PU. Once values for all pixel positions in the PU have been calculated, the intra-prediction module 46 may calculate an error value for the prediction mode based on pixel differences between the PU and the received block to be encoded. The intra-prediction module 46 may continue testing intra-prediction modes until an intra-prediction mode that yields an acceptable error value is discovered. The intra-prediction module 46 may then send the PU to the summer 50.

[0108] The video encoder 20 forms a residual block by subtracting the prediction data calculated by the motion compensation unit 44 or the intra-prediction module 46 from the original video block being coded. The summer 50 represents the component or components that perform this subtraction operation. The residual block may correspond to a two-dimensional matrix of pixel difference values, where the number of values in the residual block is the same as the number of pixels in the PU corresponding to the residual block. The values in the residual block may correspond to the differences, i.e., error, between values of co-located pixels in the PU and in the original block to be coded. The differences may be chroma or luma differences depending on the type of block that is coded.

[0109] The transform module 52 may form one or more transform units (TUs) from the residual block. The transform module 52 applies a transform, such as a discrete cosine transform (DCT), a directional transform, or a conceptually similar transform, to the TU, producing a video block comprising transform coefficients. The transform module 52 may send the resulting transform coefficients to the quantization unit 54. The quantization unit 54 may then quantize the transform coefficients. The entropy encoding unit 56 may then perform a scan of the quantized transform coefficients in the matrix according to a specified scan order. This disclosure describes the entropy encoding unit 56 as performing the scan. However, it should be understood that, in other examples, other processing units, such as the quantization unit 54, could perform the scan.

[0110] As mentioned above, scanning of the transform coefficients may involve two scans. One scan identifies which of the coefficients are significant (i.e., nonzero) to form the significance map and another scan codes the levels of the transform coefficients. In one example, this disclosure proposes that the scan order used to code

coefficient levels in a block be the same as the scan order used to code significant coefficients in the significance map for the block. In HEVC, the block may be a transform unit. As used herein, the term scan order may refer to either the direction of the scan and/or the pattern of the scan. As such, the scans for the significance map and the coefficient levels may be the same in scan pattern and/or scan direction. That is, as one example, if the scan order used to form the significance map is a horizontal scan pattern in the forward direction, then the scan order for coefficient levels should also be a horizontal scan pattern in the forward direction. Likewise, as another example, if the scan order for the significance map is a vertical scan pattern in the inverse direction, then the scan order for coefficient levels should also be a vertical scan pattern in the inverse direction. The same may apply for diagonal, zig-zag or other scan patterns.

[0111] FIG. 6 shows examples of inverse scan orders for a block of transform coefficients, i.e., a transform block. The transform block may be formed using a transform such as, for example, a discrete cosine transform (DCT). Note that each of the inverse diagonal pattern 9, inverse zig-zag pattern 29, the inverse vertical pattern 31, and the inverse horizontal pattern 33 proceed from the higher frequency coefficients in the lower right corner of the transform block to lower frequency coefficients in the upper left corner of the transform block. Hence, one aspect of the disclosure presents a unified scan order for coding of the significance map and coding of coefficient levels. The proposed technique applies the scan order used for the significance map to the scan order used for coefficient level coding. In general, horizontal, vertical and diagonal scan patterns have been shown to work well, thus reducing the need for additional scan patterns. However, the general techniques of this disclosure are applicable for use with any scan pattern.

[0112] According to another aspect, this disclosure proposes that the significance scan be performed as an inverse scan, from the last significant coefficient in the transform unit to the first coefficient (i.e., the DC coefficient) in the transform unit. Examples of inverse scan orders are shown in FIG. 6. In particular, the significance scan proceeds from the last significant coefficient at a higher frequency position to significant coefficients at lower frequency positions, and ultimately to the DC coefficient position.

[0113] To facilitate the inverse scan, techniques for identifying the last significant coefficient may be used. A process for identifying the last significant coefficient is described in J. Sole, R. Joshi, I.-S. Chong, M. Coban, M. Karczewicz, "Parallel Context Processing for the significance map in high coding efficiency," JCTVC-D262, 4th JCT-

VC Meeting, Daegu, KR, January 2011, and in U.S. provisional patent application no. 61/419,740, filed December 3, 2010, to Joel Sole Rojals et al., entitled "Encoding of the position of the last significant transform coefficient in video coding." Once the last significant coefficient in the block is identified, then an inverse scan order can be applied for both significance map and coefficient level.

[0114] This disclosure also proposes that the significance scan and coefficient level scan are not inverse and forward, respectively, but instead have the same scan direction and, more particularly, only one direction in a block. Specifically, it is proposed that the significance scan and coefficient level scan both use an inverse scan order from the last significant coefficient in a transform unit to the first coefficient. Hence, the significance scan is performed backwards (inverse scan relative to the currently proposed scan for HEVC) from the last significant coefficient to the first coefficient (the DC coefficient). This aspect of the disclosure presents a unified, uni-directional scan order for coding of the significance map and coding of coefficient levels. In particular, the unified, uni-directional scan order may be a unified inverse scan order. The scan orders for significance and coefficient level scans according to a unified inverse scan pattern could be inverse diagonal, inverse zig-zag, inverse horizontal or inverse vertical as shown in FIG. 6. However, any scan pattern could be used.

[0115] Instead of defining sets of coefficients in two-dimensional sub-blocks as shown in FIG. 3 for the goal of CABAC context derivation, this disclosure proposes defining sets of coefficients as several coefficients that are consecutively scanned according to the scan order. In particular, each set of coefficients may comprise consecutive coefficients in the scan order over the entire block. Any size of the set can be considered, although a size of 16 coefficients in a scan set has been found to work well. The set size might be fixed or adaptive. This definition allows for sets to be 2-D blocks (if a sub-block scanning method is used), rectangles (if horizontal or vertical scans are used), or diagonal shaped (if zig-zag or diagonal scans are used). The diagonal shaped sets of coefficients may be part of a diagonal shape, consecutive diagonal shapes, or parts of consecutive diagonal shapes.

[0116] FIGS. 7-9 show examples of coefficients arranged into 16 coefficient sub-sets according to specific scan orders beyond being arranged in fixed 4x4 blocks. FIG. 7 depicts a 16 coefficient sub-set 51 that consists of the first 16 coefficients in an inverse diagonal scan order. The next sub-set, in this example, would simply consist of the next 16 consecutive coefficients along the inverse diagonal scan order. Similarly, FIG. 8

depicts the 16 coefficient sub-set 53 for the first 16 coefficients an inverse horizontal scan order. FIG. 9 depicts the 16 coefficient sub-set 55 for the first 16 coefficients in an inverse vertical scan order.

[0117] This technique is compatible with a scan order for the coefficient levels that is the same as the scan order for the significance map. In this case, there is no need for a different (and sometimes cumbersome) scan order for the coefficients levels, such as that shown in FIG. 3. The coefficient level scan may be formed, like the significance map scan presently proposed for HEVC, as a forward scan that proceeds from the position of the last significant coefficient in a transform unit to the DC coefficient position.

[0118] As currently proposed in HEVC, for entropy coding using CABAC, transform coefficients are encoded in the following way. First, there is one pass (in the significance map scan order) on the full transform unit to encode the significance map. Then, there are three passes (in the coefficient level scan order) to encode bin 1 of the level (1st pass), the rest of the coefficient level (2nd pass) and the sign of the coefficient level (3rd pass). These three passes for coefficient level coding are not done for the full transform unit. Instead, each pass is done in 4x4 sub-blocks, as shown in FIG. 3. When the three passes have been completed in one sub-block, the next sub-block is processed by sequentially performing the same three encoding passes. This approach facilitates the parallelization of the encoding.

[0119] As described above, this disclosure proposes to scan transform coefficients in a more harmonized manner, such that the scan order for the coefficient levels is the same as the scan order of significant coefficients to form the significance map. In addition, it is proposed that the scans for coefficient level and significant coefficients be performed in an inverse direction that proceeds from the last significant coefficient in the block to the first coefficient (the DC component) in the block. This inverse scan is the opposite of the scan used for significant coefficients according to HEVC, as currently proposed.

[0120] As previously described with reference to FIGS. 7-9, this disclosure further proposes that the contexts for the coefficient levels (including the significance map) are divided into sub-sets. That is, a context is determined for each sub-set of coefficients. Hence, in this example, the same context is not necessarily used for an entire scan of coefficients. Instead, different sub-sets of coefficients within a transform block may have different contexts that are individually determined for each sub-set. Each sub-set may comprise a one-dimensional array of consecutively scanned coefficients in the scan

order. Therefore, the coefficient level scan goes from the last significant coefficient to the first coefficient (DC component), where the scan is conceptually partitioned in different sub-sets of consecutively scanned coefficients according to the scan order. For example, each sub-set may include n consecutively scanned coefficients, for a particular scan order. Grouping coefficients in sub-sets according to their scan order may provide for a better correlation between coefficients, and thus more efficient entropy coding.

[0121] This disclosure further proposes to increase the parallelization of the CABAC-based entropy coding of the transform coefficients by extending the concept of the several passes of the coefficient level to include an additional pass for the significance map. Thus, an example with four passes may include: (1) coding of significant coefficient flag values for the transform coefficients, e.g., to form the significance map, (2) coding of bin 1 of level values for transform coefficients, (3) coding of remaining bins of the coefficient level values, and (4) coding of the signs of the coefficient levels, all in the same scan order. Using the techniques described in this disclosure, the four-pass coding outlined above can be facilitated. That is, scanning significant coefficients and levels for transform coefficients in the same scan order, where the scan order proceeds in an inverse direction from a high frequency coefficient to a low frequency coefficient, supports the performance of the several-pass coding technique described above.

[0122] In another example, a five pass scan technique may include: (1) coding of significant coefficient flag values for the transform coefficients, e.g., to form the significance map, (2) coding of bin 1 of level values for transform coefficients, (3) coding of bin 2 of level values for transform coefficients, (4) coding the signs of the coefficient levels (e.g., in bypass mode), and (5) coding of remaining bins of the coefficient level values (e.g., in bypass mode), all passes using the same scan order.

[0123] An example with fewer passes may also be employed. For example, a two pass scan where level and sign information are processed in parallel may include: (1) coding regular pass bins in pass (e.g., significance, bin 1 level, and bin 2 level), and (2) coding bypass bins in another (e.g., remaining levels and sign), each pass using the same scan order. Regular bins are bins encoded with CABAC using updated context determined by context derivation criteria. For example, as will be explained in more detail below, the context derivation criteria may include the coded level information of causal neighbor coefficient relative to a current transform coefficient. Bypass bins are bins encoded with CABAC having a fixed context.

[0124] The examples the several scan passes described above may be generalized as including a first scan pass of the first portion of the coefficients levels, wherein the first portion includes the significance pass, and a second scan pass of a second portion of the coefficient levels.

[0125] In each of the examples given above, the passes may be performed sequentially in each sub-set. Although the use of one-dimensional sub-sets comprising consecutively scanned coefficients may be desirable, the several pass method also may be applied to sub-blocks, such as 4x4 sub-blocks. Example two-pass and four-pass processes for consecutively scanned sub-sets are outlined in more detail below.

[0126] In a simplified two-pass process, for each sub-set of a transform unit, the first pass codes the significance of the coefficients in the sub-set following the scan order, and the second pass codes the coefficient level of the coefficients in the sub-set following the same scan order. The scanning order may be characterized by a scan direction (forward or inverse) and a scan pattern (e.g., horizontal, vertical, or diagonal). The algorithm may be more amenable to parallel processing if both passes in each sub-set follow the same scan order, as described above.

[0127] In a more refined four-pass process, for each sub-set of a transform unit, the first pass codes the significance of the coefficients in the sub-set, the second pass codes bin 1 of the coefficient level of the coefficients in the sub-set, the third pass codes the remaining bins of the coefficient level of the coefficients in the sub-set, and the fourth pass codes the sign of the coefficient level of the coefficients in the sub-set. Again, to be more amenable to parallel processing, all passes in each sub-set should have the same scan order. As is described above, a scan order with an inverse direction has been shown to work well. It should be noted that the fourth pass (i.e., the coding of the sign of the coefficient levels) may be done immediately after the first pass (i.e., the coding of the significance map) or right before the remaining values of the coefficient level pass.

[0128] For some transform sizes, the sub-set can be the entire transform unit. In this case, there is a single sub-set corresponding to all of the significant coefficients for an entire transform unit, and the significance scan and level scan proceed in the same scan order. In this case, instead of limited number of n (e.g., n=16) coefficients in a sub-set, the sub-set could be a single sub-set for a transform unit, wherein the single subset includes all significant coefficients.

[0129] Returning to FIG. 5, once the transform coefficients are scanned, the entropy encoding unit 56 may apply entropy coding such as CAVLC or CABAC to the

coefficients. In addition, the entropy encoding unit 56 may encode motion vector (MV) information and any of a variety of syntax elements useful in decoding the video data at the video decoder 30. The syntax elements may include a significance map with significant coefficient flags that indicate whether particular coefficients are significant (e.g., non-zero) and a last significant coefficient flag that indicates whether a particular coefficient is the last significant coefficient. The video decoder 30 may use these syntax elements to reconstruct the encoded video data. Following the entropy coding by the entropy encoding unit 56, the resulting encoded video may be transmitted to another device, such as the video decoder 30, or archived for later transmission or retrieval.

[0130] To entropy encode the syntax elements, the entropy encoding unit 56 may perform CABAC and select context models based on, for example, the number of significant coefficients in the previously scanned *N* coefficients, where *N* is an integer value that may be related to the size of the block being scanned. The entropy encoding unit 56 may also select the context model based on a prediction mode used to calculate residual data that was transformed into the block of transform coefficients, and a type of transform used to transform the residual data into the block of transform coefficients. When the corresponding prediction data was predicted using an intra-prediction mode, the entropy encoding unit 56 may further base the selection of the context model on the direction of the intra-prediction mode.

[0131] Further, according to another aspect of this disclosure, it is proposed that the contexts for CABAC be divided into sub-sets of coefficients (e.g., the sub-sets shown in FIGS. 7-9. It is proposed that each sub-set is composed of consecutive coefficients in the scanning order over the entire block. Any size of the sub-set can be considered, although a size of 16 coefficients in a scan sub-set has been found to work well. In this example, a sub-set may be 16 consecutive coefficients in the scan order, which may be in any scan pattern, including sub-block, diagonal, zig-zag, horizontal, and vertical scan patterns. According to this proposal, the coefficient level scan proceeds from the last significant coefficient in a block. Therefore, the coefficient level scan goes from the last significant coefficient to the first coefficient (DC component) in the block, where the scan is conceptually partitioned in different sub-sets of coefficients in order to derive the contexts to apply. For example, the scan is arranged in sub-sets of n consecutive coefficients in the scan order. The last significant coefficient is the first significant coefficient encountered in an inverse scan from the highest frequency coefficient of the

block (typically found near the lower right corner of the block) toward the DC coefficient of the block (upper left corner of the block).

[0132] In another aspect of the disclosure, it is proposed that CABAC context derivation criteria be harmonized for all block sizes. In other words, instead of having different context derivations based on block size as discussed above, every block size will rely on the same derivation of CABAC contexts. In this way, there is no need to take into account the specific block size in order to derive the CABAC context for the block. Context derivation is also the same for both significance coding and coefficient level coding.

[0133] It is also proposed that CABAC context sets depend on whether the sub-set is sub-set 0 (defined as the sub-set with the coefficients for the lowest frequencies, i.e., containing the DC coefficient and adjacent low frequency coefficients) or not (i.e., the context derivation criteria). See Tables 3a and 3b below.

Context Set		
0	Lowest freq	0 LargerT1 in previous sub-set
1	Lowest freq	1 LargerT1 in previous sub-set
2	Lowest freq	>1 LargerT1 in previous sub-set
3	Higher freq	0 LargerT1 in previous sub-set
4	Higher freq	1 LargerT1 in previous sub-set
5	Higher freq	>1 LargerT1 in previous sub-set

Table 3a-Context sets table. To be compared with Table 2. There is a dependency on the sub-set, whether it is sub-set 0 (lowest frequencies) or not.

[0134] Per Table 3a above, sets 0-2 of context models are used for the lowest frequency scan sub-set (i.e., set of n consecutive coefficients) if, respectively, there are zero coefficients larger than one in the previously coded sub-set, there is one coefficient larger than one in the previously coded sub-set, or there are more than one coefficient larger than one in the previously coded sub-set. Sets 3-5 of context models are used for all sub-sets higher than the lowest frequency sub-set if, respectively, there are there are

zero coefficients larger than one in the previously coded sub-set, there is one coefficient larger than one in the previously coded sub-set, or there are more than one coefficient larger than one in the previously coded sub-set.

Context Set		
0	Lowest freq	0 LargerT1 in previous sub-set
1	Lowest freq	1-3 LargerT1 in previous sub-set
2	Lowest freq	>3 LargerT1 in previous sub-set
3	Higher freq	0 LargerT1 in previous sub-set
4	Higher freq	1-3 LargerT1 in previous sub-set
5	Higher freq	>3 LargerT1 in previous sub-set

Table 3b - Context set table.

[0135] Table 3b shows a context set table that has showed good performance as it accounts for a more precise count of the number of larger than one coefficients in the previous sub-set. Table 3b may be used as an alternative to Table 3a above.

[0136] Table 3c shows a simplified context set table with context derivation criteria that may also be alternatively used.

Context Set		
0	Lowest freq	0 LargerT1 in previous sub-set
1	Lowest freq	1 LargerT1 in previous sub-set
2	Higher freq	0 LargerT1 in previous sub-set
3	Higher freq	1 LargerT1 in previous sub-set

Table 3c - Context set table.

[0137] In addition, a sub-set containing the last significant coefficient in the transform unit may utilize a unique context set.

[0138] This disclosure also proposes that the context for a sub-set still depends on the number of coefficients larger than 1 in previous sub-sets. For example, if the number of coefficients in previous sub-sets is a sliding window, let this number be uiNumOne. Once this value is checked to decide the context for the current sub-scan set, then the value is not set to zero. Instead, this value is normalized (e.g., use uiNumOne = uiNumOne/4 which is equivalent to uiNumOne >>= 2, or uiNumOne = uiNumOne/2 which is equivalent to uiNumOne >>= 1). By doing this, values of sub-sets prior to the immediately previous sub-set can still be considered, but given smaller weight in the CABAC context decision for the currently coded sub-set. In particular, the CABAC context decision for a given sub-set takes into account not only the number of coefficients greater than one in the immediately preceding sub-set, but also a weighted number of coefficients greater than one in previously coded sub-sets.

[0139] Additionally, the context set can depend on the following: (1) the number of significant coefficients in the currently scanned sub-set,(2) whether the current sub-set is the last sub-set with a significant coefficient (i.e., using the inverse scan order, this refers to whether the sub-set is the first scanned for the coefficients levels or not). Additionally, the context model for a coefficient level can depend on whether the current coefficient is the last coefficient.

[0140] A highly adaptive context selection approach has previously been proposed for significance map coding of 16x16 and 32x32 blocks of transform coefficients in HEVC. It should be noted that this context selection approach may be extended to all block sizes. As shown in FIG. 10, this approach divides a 16x16 block into four regions, where each coefficient in the lower frequency region 41 (the four coefficients at the top left corner in x, y coordinate positions [0,0], [0,1], [1,0], [1,1] in the example of a 16x16 block, where [0,0] indicates the upper left corner, DC coefficient) has its own context, the coefficients in the top region 37 (coefficients in the top row from x, y coordinate positions [2,0] to [15,0] in the example of a 16x16 block) share 3 contexts, the coefficients in the left region 35 (coefficients in the left column from x, y coordinate positions [0,2] to [0,15] in the example of a 16x16 block) share another 3 contexts, and the coefficients in the remaining region 39 (remaining coefficients in the 16x16 block) share 5 contexts. Context selection for transform coefficient X in region 39, as an example, is based on the sum of the significance of a maximum of the 5 transform coefficients B, E, F, H, and I. Since X is independent of other positions on the same diagonal line of X along the scan direction (in this example a zig-zag or diagonal scan

pattern), the context of the significance of the transform coefficients along a diagonal line in scan order could be computed in parallel from the previous diagonal lines in scan order.

[0141] The proposed contexts for the significance map, as shown in FIG. 10, are only valid if the scan order is forward because the context becomes non-causal at the decoder if an inverse scan is used. That is, the decoder has not yet decoded coefficients B, E, F, H and I as shown in FIG. 10 if the inverse scan is used. As a result, the bitstream is not decodable.

[0142] However, this disclosure proposes the use of an inverse scan direction. As such, the significance map has relevant correlation among coefficients when the scan order is in an inverse direction, as shown in FIG. 6. Therefore, using an inverse scan for the significance map, as described above, offers desirable coding efficiency. Also, the use of an inverse scan for the significance map serves to harmonize the scanning used for coding of coefficient level and the significance map. To support inverse scanning of the significant coefficients, the contexts need to be changed so that they are compatible with an inverse scan. It is proposed that coding of significant coefficients make use of contexts that are causal with respect to the inverse scan.

[0143] This disclosure further proposes, in one example, a technique for significance map coding that makes use of the contexts depicted in FIG. 11. Each coefficient in the lower frequency region 43 (the three coefficients at the top left corner in x, y coordinate positions [0,0], [0,1], [1,0] in the example of a 16x16 block, where [0,0] indicates the upper left corner, DC coefficient) has its own context derivation. The coefficients in the top region 45 (coefficients in the top row from x, y coordinate positions [2,0] to [15,0] in the example of a 16x16 block) have a context dependent on the significance of the two previous coefficients in the top region 45 (e.g., two coefficients immediately to the right of the coefficient to be coded, where such coefficients are causal neighbors for decoding purposes given the inverse scan). The coefficients in the left region 47 (coefficients in the left column from x, y coordinate positions [0,2] to [0,15] in the example of a 16x16 block) have a context dependent on the significance of the two previous coefficients (e.g., two coefficients immediately below the coefficient to be coded, where such coefficients are causal neighbors for decoding purposes given the inverse scan orientation). Note that these contexts in the top region 45 and the left region 47 in FIG. 11 are the inverse of the contexts shown in FIG. 10 (e.g., where the coefficients in the top region 37 have a context dependent on coefficients to the left and

the coefficients in the left region 35 have a context dependent on coefficients above). Returning to FIG. 11, the contexts for the coefficients in the remaining region 49 (i.e., the remaining coefficients outside of lower frequency region 43, top region 45, and left region 47) depend on the sum (or any other function) of the significance of the coefficients in the positions marked with I, H, F, E, and B.

[0144] In another example, the coefficients in top region 45 and left region 47 may use exactly the same context derivation as the coefficients in region 49. In the inverse scan, that is possible because the neighboring positions marked with I, H, F, E, and B are available for the coefficients in the top region 45 and the left region 47. At the end of the rows/columns, the positions for causal coefficients I, H, F, E and B might be outside the block. In that case, it is assumed that the value of such coefficients is zero (i.e., non-significant).

[0145] There are many options in choosing the contexts. The basic idea is to use the significance of coefficients that have already been coded according to the scan order. In the example shown in FIG. 10, the context of the coefficient at position X is derived based on the sum of the significance of coefficients at positions B, E, F, H and I. These context coefficients come before the current coefficient in the inverse scan order proposed in this disclosure for the significance map. The contexts that were causal in the forward scan become non-causal (not available) in the inverse scan order. A way to address this problem is to mirror the contexts of the conventional case in FIG. 10 to those shown in FIG. 11 for the inverse scan. For a significance scan that proceeds in an inverse direction from a last significant coefficient to a DC coefficient position, the context neighborhood for coefficient X is composed of coefficients B, E, F, H, I, which are associated with higher frequency positions, relative to the position of coefficient X, and which have already been processed by the encoder or decoder, in the inverse scan, prior to coding of coefficient X.

[0146] As discussed above, contexts and context models illustrated in Tables 1 and 2 attempt to exploit the local correlation of coefficient levels among 4x4 sub-blocks. However, the dependency might be too far. That is to say, there may be a low dependency between coefficients which are separated from one another by several coefficients, for example, from one sub-block to another. Also, within each sub-block, the dependency between coefficients might be weak. This disclosure describes techniques for addressing these issues by the creation of a set of contexts for the coefficient levels that exploit a more local context neighborhood.

[0147] This disclosure proposes to use a local neighborhood for the derivation of the context of the transform coefficient levels, e.g., in video coding according to HEVC or other standards. This neighborhood is composed of coefficients already encoded (or decoded) that have high correlation with the level of the current coefficient. The coefficients may spatially neighbor the coefficient to be coded, and may include both coefficients that bound the coefficient to be coded and other nearby coefficients, such as shown in FIG. 11 or FIG. 13. Notably, the coefficients used for context derivation are not constrained to a sub-block or previous sub-block. Instead, the local neighborhood may comprise coefficients that are spatially located close to the coefficient to be coded, but would not necessarily reside in the same sub-block as the coefficient to be coded, or in the same sub-block as one another, if the coefficients were arranged in sub-blocks. Rather than relying on coefficients located in a fixed sub-block, this disclosure proposes using neighboring coefficients that are available (i.e., have already been coded) given the specific scan order used.

[0148] Different CABAC context sets may be specified for different sub-sets of coefficients, e.g., based on previously coded sub-sets of coefficients. Within a given sub-set of coefficients, contexts are derived based on a local neighborhood of coefficients, sometimes called a context neighborhood. In accordance with this disclosure, an example of a context neighborhood is shown in FIG. 12. The coefficients in the context neighborhood may be spatially located near the coefficient to be coded.

[0149] As shown in FIG. 12, for a forward scan, the context of the level for the transform coefficient X depends on the values of the coefficients B, E, F, H, and I. In a forward scan, coefficients B, E, F, H, and I are associated with lower frequency positions relative to the position and coefficient X, and which have already been processed by the encoder or decoder prior to coding of coefficient X.

[0150] For encoding bin 1 for CABAC, the context depends on the sum of the number of significant coefficients in this context neighborhood (i.e., in this example, coefficients B, E, F, H and I). If a coefficient in the context neighborhood falls out of the block, i.e., due to data loss, it can be considered that the value is 0 for purposes of determining the context of coefficient X. For encoding the rest of the bins for CABAC, the context depends on the sum of the number of coefficients in the neighborhood that are equal to 1 as well as on the sum of the number of coefficients in the neighborhood that are larger than 1. In another example, the context for bin 1 may depend on the sum of the bin 1 values of the coefficients in the local context neighborhood. In another

example, the context for bin 1 may depend on a combination of the sum of the significance coefficients and bin 1 values in this context neighborhood.

[0151] There are many possibilities for the selection of a context neighborhood. However, the context neighborhood should be composed of coefficients such that encoder and decoder both have access to the same information. In particular, the coefficients B, F, E, I, and H in the neighborhood should be causal neighbors in the sense that they have been previously encoded or decoded and are available for reference in determining the context for coefficient X.

[0152] The contexts described above with reference to FIG. 12 are one of many possibilities. Such contexts can be applied to any of the three scans currently proposed for use in HEVC: diagonal, horizontal and vertical. This disclosure proposes that the context neighborhood used to derive the context for the coefficient level may be the same as the context neighborhood used for deriving contexts for the significance map. For example, the context neighborhood used to derive the context for the coefficient level may be a local neighborhood, as is the case for the coding of the significance map.

[0153] As described in more detail above, this disclosure proposes the use of an inverse scan order for the scanning of significant coefficients to form the significance map. The inverse scan order may be an inverse zig-zag pattern, vertical pattern or horizontal pattern as shown in FIG. 6. If the scan order for the coefficient level scan is also in an inverse pattern, then the context neighborhood shown in FIG. 12 would become non-causal. This disclosure proposes to reverse the position of the context neighborhood so that they are causal with reference to an inverse scan order. FIG. 13 shows an example of a context neighborhood for an inverse scan order.

[0154] As shown in FIG. 13, for a level scan that proceeds in an inverse direction from a last significant coefficient to a DC coefficient position, the context neighborhood for coefficient X is composed of coefficients B, E, F, H, and I, which are associated with higher frequency positions relative to the position of coefficient X. Given the inverse scan, coefficients B, E, F, H, and I have already been processed by the encoder or decoder, prior to coding of coefficient X, and are therefore causal in the sense that they are available. Similarly, this context neighborhood can be applied to the coefficient levels.

[0155] This disclosure further proposes, in one example, another technique for significance map coding that makes use of the contexts selected to support an inverse scan. As discussed above, a highly adaptive context selection approach has been

proposed for HEVC for significance map coding of 16x16 and 32x32 blocks of transform coefficients. For instance, as was described in reference to FIG. 10 above, this approach divides a 16x16 block into four regions, where each position in the region 41 has its own set of contexts, region 37 has contexts, region 35 has another 3 contexts, and region 39 has 5 contexts. Context selection for transform coefficient X, as an example, is based on the sum of the significance of a maximum of the 5 positions B, E, F, H, I. Since X is independent of other positions on the same diagonal line of X along the scan direction, the context of the significance of the transform coefficients along a diagonal line in scan order could be computed in parallel from the previous diagonal lines in scan order.

[0156] The current HEVC approach for context derivations has several drawbacks. One issue is the number of contexts per block. Having more contexts implies more memory and more processing each time the contexts are refreshed. Therefore, it would be beneficial to have an algorithm that has few contexts and also, few ways to generate the contexts (e.g., less than the four ways, i.e., four patterns, in the previous example).

[0157] One way to address such drawbacks is the coding of the significance map in inverse order, that is, from the last significant coefficient (higher frequency) to the DC component (lowest frequency). A consequence of this process in inverse order is that the contexts for the forward scanning are no longer valid. The techniques described above include a method for determining the contexts for context adaptive binary arithmetic coding (CABAC) of the information indicating a current one of the significant coefficients based on previously coded significant coefficients in the inverse scan direction. In an example of inverse zig-zag scan, the previously coded significant coefficients reside at positions to the right of a scan line on which the current of the significant coefficients resides.

[0158] The context generation might be different for different positions of the transform blocks based on, at least, the distance from the boundaries and the distance from the DC component. In the example technique described above, it was proposed that the significance map coding makes use of the sets of contexts depicted in FIG. 11.

[0159] This disclosure proposes a set of contexts for inverse significance map scan that may lead to higher performance through the reduction of the number of contexts per block. Referring back to FIG. 11, a reduction in the number of context per block may be accomplished by allowing the left region 47 and top region 45 to use the same context derivation as the remaining region 49. In the inverse scan that is possible

because the neighboring positions marked with I, H, F, E, and B are available for the coefficients at the regions 47 and 45.

[0160] FIG. 14 shows an example of a context derivation according to this example. In this example there are only two context regions: a low frequency region 57 for the DC coefficient and the remaining region 59 for all other coefficients. As such, this example proposes just two ways to derive the context. In the low frequency region 57 (the DC coefficient at x, y position [0,0]), the context is derived based on the position, i.e., the DC coefficient has a context of its own. In the remaining region 57, the context is derived based on the significance of the neighbor coefficients in the local neighborhood for each coefficient to be coded. In this example, it is derived depending on the sum of the significance of the 5 neighbors denoted by I, H, F, E and B in FIG. 14.

[0161] Therefore, the number of ways to derive the context within a block is reduced from 4 to 2. Also, the number of contexts is reduced by 8 with respect to the previous example in FIG. 11 (2 lower frequency region 43 and 3 for each of upper region 45 and left region 47). In another example, the DC coefficient may use the same method as the rest of the block, so the number of ways to derive the context within a block is reduced to 1.

[0162] FIG. 15 shows an example where the current position of coefficient X causes some of the neighbor coefficients (in this case H and B) to be outside of the current block. If any of the neighbors of the current coefficient are outside the block it may be assumed that such neighbor coefficients have 0 significance (i.e., they are zero-valued and therefore non-significant). Alternatively, one or more special contexts may be specified for one or more coefficients at the lower right. In this way, higher frequency coefficients may have contexts depending on position, in a similar way as the DC coefficient. However, assuming the neighbors to be zero may provide sufficient results, particularly because the lower right coefficients will ordinarily have a low probability of having significant coefficients, or at least significant coefficients with large values.

[0163] The reduction of the number contexts in the example of FIG. 14 is good for implementation. However, it may result in a small drop in performance. This disclosure proposes a further technique to improve the performance while still reducing the number of contexts. In particular, it is proposed to have a second set of contexts that is also based on neighbor coefficients. The context derivation algorithm is exactly the same, but two sets of contexts with different probability models are used. The set of

contexts that are used depends on the position of the coefficient to be coded within the transform unit.

[0164] More specifically, increased performance has been shown when using a context model for higher frequency coefficients (e.g., lower right x, y coordinate positions of coefficients) that is different from the context model for coefficients at lower frequencies (e.g., upper left x, y coordinate positions of coefficients). One way to separate the lower frequency coefficients from the higher frequency coefficients, and thus the context model used for each, is to compute the x+y value for a coefficient, where x is the horizontal position and y is the vertical position of the coefficient. If this value is smaller than some threshold (e.g., 4 has been shown to work well), then context set 1 is used. If the value is equal to or larger than the threshold, then context set 2 is. Again, context sets 1 and 2 have different probability models.

[0165] FIG. 16 shows an example of the context regions for this example. Again, the DC coefficient at position (0,0) has its own context region 61. The lower frequency context region 63 consists of transform coefficients at an x+y position equal to or less than a threshold of 4 (not including the DC coefficient). The higher frequency context region 65 consists of transform coefficients at an x+y position greater than a threshold of 4. The threshold of 4 is used an example and may be adjusted to any number that provides for better performance. In another example, the threshold may depend on the TU size.

[0166] The context derivation for the lower frequency context region 63 and the higher frequency context region 65 is exactly the same in terms of the manner in which the neighbors are used to select context, but the employed probabilities (i.e., contexts) are different. In particular, the same criteria for context selection based on neighbors may be used, but the application of such criteria leads to selection of a different context for different coefficient positions, because different coefficient positions may be associated with different sets of contexts. In this way, the knowledge that lower and high frequency coefficients have different statistics is incorporated in the algorithm, so that different context sets for different coefficient may be used.

[0167] In other examples, the x+y function can be changed to other functions depending on the position of the coefficient. For instance, an option is to give the same set of contexts to all the coefficients with x<T && y<T, T being a threshold. FIG. 17 shows an example of a block of transform coefficients with these context regions. Again, the DC coefficient at position (0,0) may have its own context region 61. The lower

frequency context region 73 consists of all transform coefficients whose X or Y position is less than or equal to a threshold of 4 (not including the DC coefficient). The higher frequency context region consists of all transform coefficients whose X or Y position is greater than a threshold of 4. Again, the threshold of 4 is used as an example and may be adjusted to any number that provides for better performance. In one example, the threshold may depend on the TU size.

[0168] The above described techniques shown in FIGS. 16 and 17 have two sets of 5 contexts, which is still a lesser number of contexts than the number of contexts shown in FIG. 10, and exhibits higher performance. This is achieved by separating the block into different areas, and specifying different context sets for the coefficients in the different areas, but still applying the same context derivation criteria to each area.

[0169] FIG. 18 shows another example of a block of transform coefficients with context regions. In this example, the DC coefficient in region 81 and the coefficients at x, y positions (1, 0) and (0, 1), in regions 83 and 85, each have their own context. The remaining region 87 has yet another context. In a variation of the example shown in FIG. 18, regions 83 and 85 share a context.

[0170] In general, the above-described techniques may include scanning significant coefficients in a block of transform coefficients in an inverse direction from higher frequency coefficients in the block of the transform coefficients to lower frequency coefficients in the block of transform coefficients to form a significance map, and determining contexts for context adaptive binary arithmetic coding (CABAC) of the significant coefficients of the significance map based on a local neighborhood of previously scanned coefficients in the block. The contexts may be determined for each of the significant coefficients based on previously scanned transform coefficients in the local neighborhood having higher frequencies than the respective transform coefficient. In some examples, the contexts may be determined based on a sum of the number of significant coefficients in previously scanned coefficients of a context neighborhood. The local neighborhood for each of the significant coefficients to be coded may comprise a plurality of transform coefficients that spatially neighbor the respective coefficient in the block.

[0171] A context for a significant coefficient at a DC (e.g., upper left-most) position of the block of transform coefficients may be determined based on an individual context specified for the significant coefficient at the DC position. Also, a context may be determined for coefficients at a left edge and top edge of the block using criteria

substantially similar or identical to criteria used to determine context for coefficients that are not at the left edge and top edge of the block. In some examples, context for a coefficient at lower right-most position of the block may be determined using criteria that assumes that neighboring coefficients outside of the block are zero-valued coefficients. Also, in some examples, determining a context may comprise determining a context for coefficients using substantially similar or identical criteria for selection of contexts within a context set, but different context sets, based on positions of the coefficients within the block of transform coefficients.

[0172] Reference to upper, lower, right, left, and the like in this disclosure is used generally for convenience to refer to relative positions of higher frequency and lower frequency coefficients in a block of transform coefficients that is arranged, in a conventional manner, to have lower frequency coefficients toward the upper left and higher frequency coefficients toward the lower right of the block, and should not be considered limiting for cases in which the higher and lower frequency coefficients may be arranged in a different, unconventional manner.

[0173] Returning to FIG. 5, in some examples, the transform module 52 may be configured to zero out certain transform coefficients (that is, transform coefficients in certain locations). For example, the transform module 52 may be configured to zero out all transform coefficients outside of the upper-left quadrant of the TU following the transform. As another example, the entropy encoding unit 56 may be configured to zero out transform coefficients in the array following a certain position in the array. In any case, the video encoder 20 may be configured to zero out a certain portion of the transform coefficients, e.g., before or after the scan. The phrase "zero out" is used to mean setting the value of the coefficient equal to zero, but not necessarily skipping or discarding the coefficient. In some examples, this setting of coefficients to zero may be in addition to the zeroing out that may result from quantization.

[0174] The inverse quantization unit 58 and the inverse transform module 60 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual block in the pixel domain, e.g., for later use as a reference block. The motion compensation unit 44 may calculate a reference block by adding the residual block to a predictive block of one of the frames of the reference frame buffer 64. The motion compensation unit 44 may also apply one or more interpolation filters to the reconstructed residual block to calculate sub-integer pixel values for use in motion estimation. The summer 62 adds the reconstructed residual block to the motion

compensated prediction block produced by the motion compensation unit 44 to produce a reconstructed video block for storage in the reference frame buffer 64. The reconstructed video block may be used by the motion estimation unit 42 and the motion compensation unit 44 as a reference block to inter-code a block in a subsequent video frame.

[0175] FIG. 19 is a block diagram illustrating an example of an entropy encoding unit 56 for use in the video encoder of FIG. 5. FIG. 19 illustrates various functional aspects of the entropy encoding unit 56 for selecting a scan order and corresponding context set used in CABAC entropy coding. The entropy encoding unit 56 may include a scan order and context selection unit 90, a 2D-to-1D scan unit 92, an entropy encoding engine 94, and a scan order memory 96.

[0176] The scan order and context selection unit 90 selects the scan order to be used by the 2D-to-1D scan unit 92 for the significance map scan and the coefficient level scan. As discussed above, the scan order consists of both a scan pattern and a scan direction. The scan memory 96 may store instructions and/or data that defines what scan order to use for particular situations. As examples, the prediction mode of a frame or slice, the block size, the transform, or other characteristics of the video data used may be used to select the scan order. In one proposal for HEVC, each of the intra prediction modes is assigned to a particular scan order (sub-block diagonal, horizontal or vertical). The decoder parses the intra prediction mode, and determines the scan order to apply using look-up-table. Adaptive methods may be used to track the statistics of the most frequent significant coefficients. In another example, a scan may be based on the most frequently used coefficients first in scan order. As another example, the scan order and context selection unit 90 may use a predetermined scan order for all situations. As described above, the scan order and context selection unit 90 may select a scan order for both a significance map and a coefficient level scan. In accordance with techniques of this disclosure, the two scans may have the same scan order, and in particular, may both be in an inverse direction.

[0177] Based on the selected scan order, the scan order and context selection unit 90 also selects the contexts to be used for CABAC in the entropy encoding engine 94, such as the contexts described above with reference to FIG. 11 and FIGs. 13-18.

[0178] The 2D-to-1D scan unit 92 applies the selected scans to the two-dimensional array of transform coefficients. In particular, the 2D-to-1D scan unit 92 may scan the transform coefficients in sub-sets, as is described above with reference to FIGs. 7-9. In

particular, transform coefficients are scanned in a sub-set consisting of a number of consecutive coefficients according to the scan order. Such sub-sets are applicable for both the significance map scan as well as the coefficient level scan. Additionally, the 2D-to-1D scan unit 92 may perform significance map and coefficient level scans as consecutive scans and according to the same scan order. The consecutive scans may consist of several scans, as described above. In one example, the first scan is the significance map scan, the second scan is of bin one of the levels of transform coefficients in each sub-set, the third scan is of the remaining bins of the levels of transform coefficients, and the fourth scan is of the sign of the levels of transform coefficients.

[0179] The entropy encoding engine 94 applies an entropy encoding process to the scanned coefficients using the selected context from the scan order and context selection unit 90. In some examples the context used for CABAC may be predetermined for all cases, and as such, there may be no need for a process or unit to select the contexts. The entropy encoding process may be applied to the coefficients after they are fully scanned into the 1D vector, or as each coefficient is added to the 1D vector. In other examples, the coefficients are processed directly in the 2D array using the scan order. In some cases, the entropy encoding engine 94 may be configured to encode different sections of the 1D vector in parallel to promote parallelization of the entropy encoding process for increased speed and efficiency. The entropy encoding engine 94 produces a bitstream carrying the encoded video. The bitstream may be transmitted to another device or stored in a data storage archive for later retrieval. In addition to the residual transform coefficient data, the bitstream may carry motion vector data and various syntax elements useful in decoding the encoded video in the bitstream.

[0180] In addition, the entropy encoding unit 56 may provide signaling in the encoded video bitstream to indicate the scan order and/or contexts used in the CABAC process. The scan order and/or contexts may be signaled, for example, as syntax elements at various levels, such as the frame, slice, LCU, CU level or TU level. If a predetermined scan order and/or context are set, there may be no need to provide signaling in the encoded bitstream. Also, in some examples, it may be possible for the video decoder 30 to infer some of the parameter values without signaling. To permit definition of different scan orders for different TU's, it may be desirable to signal such syntax elements at the TU level, e.g., in a TU quadtree header. Although signaling in the

encoded video bitstream is described for purposes of illustration, information indicating the parameter values or function could be signaled out-of-band in side information.

[0181] In this context, signaling the scan order and/or contexts in the encoded bitstream does not require real-time transmission of such elements from the encoder to a decoder, but rather means that such syntax elements are encoded into the bitstream and are made accessible to the decoder in any fashion. This may include real-time transmission (e.g., in video conferencing) as well as storing the encoded bitstream on a computer-readable medium for future use by a decoder (e.g., in streaming, downloading, disk access, card access, DVD, Blu-ray, etc.).

[0182] It should be noted that, although shown as separate functional units for ease of illustration, the structure and functionality of the scan order and context selection unit 90, the 2D-to-1D scan unit 92, the entropy encoding engine 94, and the scan order memory 96 may be highly integrated with one another.

[0183] FIG. 20 is a block diagram illustrating an example of a video decoder 30, which decodes an encoded video sequence. In the example of FIG. 20, the video decoder 30 includes an entropy decoding unit 70, a motion compensation unit 72, an intra-prediction module 74, an inverse quantization unit 76, an inverse transformation unit 78, a reference frame buffer 82 and a summer 80. The video decoder 30 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to the video encoder 20 (FIG. 5).

[0184] The entropy decoding 70 entropy decodes the encoded video in a process that is the inverse of that used by entropy encoding unit 56 of FIG. 5. The motion compensation unit 72 may generate prediction data based on motion vectors received from the entropy decoding unit 70. The intra-prediction module 74 may generate prediction data for a current block of a current frame based on a signaled intra-prediction mode and data from previously decoded blocks of the current frame.

[0185] In some examples, the entropy decoding unit 70 (or the inverse quantization unit 76) may scan the received values using a scan mirroring the scan order used by the entropy encoding unit 56 (or the quantization unit 54) of the video encoder 20. Although the scanning of coefficients may be performed in the inverse quantization unit 76, scanning will be described for purposes of illustration as being performed by the entropy decoding unit 70. In addition, although shown as separate functional units for ease of illustration, the structure and functionality of the entropy decoding unit 70, the

inverse quantization unit 76, and other units of the video decoder 30 may be highly integrated with one another.

[0186] In accordance with the techniques of this disclosure, the video decoder 30 may scan both the significance map of the transform coefficients as well as levels of the transform coefficients according to the same scan order. That is to say, the scan order for the significance map and the level coding should have the same pattern and direction. In addition, the video encoder 30 may use a scan order for the significance map that is in the inverse direction. As another example, the video encoder 30 may use a scan order for the significance map the level coding that is harmonized in an inverse direction.

[0187] In another aspect of this disclosure, the video decoder 30 may scan transform coefficients in sub-sets. In particular, transform coefficients are scanned in a sub-set consisting of a number of consecutive coefficients according to the scan order. Such sub-sets are applicable for both the significance map scan as well as the coefficient level scan. Additionally, the video decoder 30 may perform the significance map scan and the coefficient level scan as consecutive scans according to the same scan order. In one aspect, the scan order is an inverse scan order. The consecutive scans may consist of several scans. In one example, the first scan is the significance map scan, the second scan is of bin one of the levels of transform coefficients in each sub-set, the third scan is of the remaining bins of the levels of transform coefficients, and the fourth scan is of the sign of the levels of transform coefficients.

[0188] The video decoder 30 may receive, from encoded bitstream, signaling that identifies the scan order and/or contexts used for CABAC by the video encoder 20. Additionally, or alternatively, the scan order and contexts may be inferred by the video decoder 30 based on characteristics of the coded video such as prediction mode, block size, or other characteristics. As another example, the video encoder 20 and the video decoder 30 may use predetermined scan orders and contexts for all use cases, and as such, no signaling in the encoded bitstream would be needed.

[0189] No matter how the scan order is determined, the entropy decoding unit 70 uses the inverse of the scan order to scan the 1D vector into the 2D array. The 2D array of transform coefficients produced by the entropy decoding unit 70 may be quantized and may generally match the 2D array of transform coefficients scanned by the entropy encoding unit 56 of the video encoder 20 to produce the 1D vector of transform coefficients.

[0190] The inverse quantization unit 76 inverse quantizes, i.e., de-quantizes, the quantized transform coefficients provided in the bitstream and decoded by the entropy decoding unit 70. The inverse quantization process may include a conventional process, e.g., similar to the processes proposed for HEVC or defined by the H.264 decoding standard. The inverse quantization process may include use of a quantization parameter QP calculated by the video encoder 20 for the CU to determine a degree of quantization and, likewise, a degree of inverse quantization that should be applied. The inverse quantization unit 76 may inverse quantize the transform coefficients either before or after the coefficients are converted from a 1D vector to a 2D array.

[0191] The inverse transform module 78 applies an inverse transform, e.g., an inverse DCT, an inverse integer transform, an inverse KLT, an inverse rotational transform, an inverse directional transform, or another inverse transform. In some examples, the inverse transform module 78 may determine an inverse transform based on signaling from the video encoder 20, or by inferring the transform from one or more coding characteristics such as block size, coding mode, or the like. In some examples, the inverse transform module 78 may determine a transform to apply to the current block based on a signaled transform at the root node of a quadtree for an LCU including the current block. In some examples, the inverse transform module 78 may apply a cascaded inverse transform.

[0192] The motion compensation unit 72 produces motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used for motion estimation with sub-pixel precision may be included in the syntax elements. The motion compensation unit 72 may use interpolation filters as used by the video encoder 20 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. The motion compensation unit 72 may determine the interpolation filters used by the video encoder 20 according to received syntax information and use the interpolation filters to produce predictive blocks.

[0193] The motion compensation unit 72 and the intra-prediction module 74, in an HEVC example, may use some of the syntax information (e.g., provided by a quadtree) to determine sizes of LCUs used to encode frame(s) of the encoded video sequence. The motion compensation unit 72 and the intra-prediction module 74 may also use syntax information to determine split information that describes how each CU of a frame of the encoded video sequence is split (and likewise, how sub-CUs are split). The

syntax information may also include modes indicating how each split is encoded (e.g., intra- or inter-prediction, and for intra-prediction an intra-prediction encoding mode), one or more reference frames (and/or reference lists containing identifiers for the reference frames) for each inter-encoded PU, and other information to decode the encoded video sequence.

[0194] The summer 80 combines the residual blocks with the corresponding prediction blocks generated by the motion compensation unit 72 or the intra-prediction module 74 to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in the reference frame buffer 82, which provides reference blocks for subsequent motion compensation and also produces decoded video for presentation on a display device (such as the display device 32 of FIG. 4).

[0195] As mentioned above, the techniques for scanning transform coefficients presented in this disclosure are applicable for both an encoder and a decoder. A video encoder may apply the scan order to scan transform coefficients from the two-dimensional array to the one-dimensional array, whereas a video decoder may apply the scan order, e.g., in an inverse manner to the encoder, to scan transform coefficients from the one-dimensional array to the two-dimensional array. Alternatively, a video decoder may apply the scan order to scan transform coefficients from the one-dimensional array to the two-dimensional array, and a video encoder may apply the scan order, in an inverse manner to the decoder, to scan transform coefficients from the two-dimensional array to the one-dimensional array. Hence, scanning by a coder may refer to 2D-to-1D scanning by an encoder or 1D-to-2D scanning by a decoder. In addition, scanning according to a scan order may refer to scanning in the scan order for 2D-to-1D scanning, scanning in the scan order for 1D-to-2D scanning, scanning in the inverse of the scan order for 1D-to-2D scanning, or scanning in the inverse of the scan order for 2D-to-1D scanning. Hence, the scan order may be established for scanning by the encoder or scanning by the decoder.

[0196] The video decoder 30 may operate in a manner essentially symmetrical to that of the video encoder 20. For example, the video decoder 30 may receive entropy encoded data representative of an encoded CU, including encoded PU and TU data. The video decoder 30 may inverse entropy encode the received data, forming encoded quantization coefficients. When the video encoder 20 entropy encodes data using an arithmetic coding algorithm (e.g., CABAC), the video decoder 30 may use a context model to

decode the data, which corresponds to the same context model used by the video encoder 20 to encode the data.

[0197] The video decoder 30 may also inverse scan the decoded coefficients, using an inverse scan that mirrors the scan used by the video encoder 20. To inverse scan the coefficients, the video decoder 30 selects the same scan order used by the video encoder 20, which may be stored at the decoder or signaled by the encoder in the encoded bitstream. Using this scan order, the video decoder 30 thereby forms a two-dimensional matrix from the one-dimensional vector of quantized transform coefficients resulting from the entropy decoding process. In particular, video decoder 30 inverse scans the coefficients from the one-dimensional array into a two-dimensional array according to the scan order used by video encoder 20.

[0198] Next, the video decoder 30 may inverse quantize the coefficients in the two-dimensional matrix produced by the inverse scan performed according to the scan order. The video decoder 30 may then apply one or more inverse transforms to the two-dimensional matrix. The inverse transforms may correspond to the transforms applied by the video encoder 20. The video decoder 30 may determine the inverse transforms to apply based on, for example, information signaled at the root of a quadtree corresponding to the CU currently being decoded, or by reference to other information indicative of the appropriate inverse transforms. Upon application of the inverse transform(s), the video decoder 30 recovers the residual video data in the pixel domain and applies intra-predictive or inter-predictive decoding, as applicable, to reconstruct the original video data.

[0199] FIG. 21 is a block diagram illustrating an example of an entropy decoding unit 70 for use in the video decoder of FIG. 20. FIG. 21 illustrates various functional aspects of the entropy decoding unit 70 for selecting a scan order and contexts used for CABAC decoding in a video decoding process. As shown in FIG. 21, the entropy encoding unit 70 may include a scan order and context selection unit 100, a 1D-to-2D scan unit 102, an entropy decoding engine 104, and a scan order memory 106.

[0200] The entropy decoding engine 104 entropy decodes encoded video transmitted to the video decoder 30 or retrieved by the video decoder 30 from a storage device. For example, the entropy decoding engine 104 may apply an entropy decoding process, e.g., CAVLC, CABAC or another process, to the bitstream carrying the encoded video to recover the 1D vector of transform coefficients. In addition to the residual transform coefficient data, the entropy decoding engine 104 may apply entropy decoding to

reproduce motion vector data and various syntax elements useful in decoding the encoded video in the bitstream. The entropy decoding engine 104 may determine which entropy decoding process, e.g., CAVLC, CABAC or another process, to select based on signaling in the encoded video bitstream or by inferring the appropriate process from other information in the bitstream.

[0201] In accordance with the techniques of this disclosure, the entropy decoding engine 104 may entropy decode the encoded video using CABAC in accordance with two different context regions. The scan order and context selection unit 100 may provide the context derivation to the entropy decoding engine 104. According to examples of this disclosure, the context derivation for a first context region depends on the position of the transform coefficients while the context derivation for the second region depends on causal neighbors of the transform coefficients. In another example, the second context region can use two different context models depending on the location of the transform coefficients.

[0202] The scan order and context selection unit 100 may also determine the scan order, and/or an indication of the scan order, based on signaling in the encoded video bitstream. For example, entropy decoding unit 70 may receive syntax elements that explicitly signal the scan order. Again, although signaling in the encoded video bitstream is described for purposes of illustration, the scan order could be received by entropy decoding unit 70 as out-of-band in side information. Also, in some examples, it may be possible for the scan order and context selection unit 100 to infer the scan order without signaling. The scan order may be based on prediction mode, block size, transform, or other characteristics of the encoded video. Like the memory 96 of FIG. 19, the memory 106 of FIG. 21 may store instructions and/or data defining the scan order.

[0203] The 1D-to-2D scan unit 102 receives the scan order from scan order and context selection unit 100 and applies the scan order, either directly or in an inverse manner, to control the scanning of coefficients. In accordance with techniques of this disclosure, the same scan order may be used for both the significance map scan and the coefficient level. In another aspect of the disclosure, the significance map scan may be in an inverse direction. In another aspect of the disclosure, both the significance map scan and the coefficient level scan may be in the inverse direction.

[0204] According to another aspect of the disclosure, the 1D-to-2D scan unit 102 may scan the one-dimensional array of transform coefficients into one or more sub-sets of

transform coefficients, coding significance of transform coefficients in each sub-set, and coding levels of transform coefficients in each sub-set. In another aspect of the disclosure, the significance map and the coefficient level scan are performed in consecutive scans according to the same scan order. In one aspect, the scan order is an inverse scan order. The consecutive scans may consist of several scans where the first scan is the significance map scan, the second scan is of bin one of the levels of transform coefficients in each sub-set, the third scan is of the remaining bins of the levels of transform coefficients, and the fourth scan is of the sign of the levels of transform coefficients.

[0205] On the encoder side, coding of the transform coefficients may comprise encoding the transform coefficients according to the scan order to form the one-dimensional array of the transform coefficients. On the decoder side, coding the transform coefficient may comprise decoding the transform coefficients according to the scan order to reconstruct a two-dimensional array of the transform coefficients in the transform block.

[0206] It should be noted that, although shown as separate functional units for ease of illustration, the structure and functionality of the scan order and context selection unit 100, the 1D-to-2D scan unit 102, the entropy decoding engine 104, and the scan order memory 106 may be highly integrated with one another.

[0207] FIG. 22 is a flowchart illustrating an example process for significance map and coefficients level scanning with a harmonized scan order. A method of coding a plurality of transform coefficients associated with residual video data in a video coding process is proposed. The method may be performed by a video coder, such as video encoder 20 or video decoder 30 of FIG. 4. The video coder may be configured to select a scan order (120). The scan order may be selected based on prediction mode, block size, transform, or other video characteristics. In addition, the scan order may be a default scan order. The scan order defines both a scan pattern and a scan direction. In one example, the scan direction is an inverse scan direction proceeding from higher frequency coefficients in the plurality of transform coefficients to lower frequency coefficients in the plurality of transform coefficients. The scan pattern may include one of a zig-zag pattern, a diagonal pattern, a horizontal pattern or a vertical pattern.

[0208] The video coder may be further configured to code information indicating significant coefficients for the plurality of transform coefficients according to a scan order (122) and determine contexts for coding the levels of the significant coefficients

for a plurality of sub-sets of the significant coefficients, wherein each of the plurality of sub-sets comprises one or more significant coefficients scanned according to the scan order (124). The video coder also codes information indicating levels of the plurality of transform coefficients according to the scan order (126). The sub-sets may be of different sizes. It should be noted that steps 122, 124 and 126 may be interleaved, as the determination of contexts for level information depends on the previously coded neighbor coefficients.

[0209] FIG. 23 is a flowchart illustrating another example process for significance map and coefficients level scanning and CABAC context derivation. The method of FIG. 23 slightly differs from that shown in FIG. 22, as contexts of blocks of different sizes may use the same context derivation criteria. As one example, the video coder may derive a first context for a first block of the transform coefficients, the first block having a first size, according to context derivation criteria, and derive a second context for a second block of the transform coefficients, the second block having a second, different size, according to the same context derivation criteria as the first block (123). Like FIG. 22, steps 122, 123 and 126 may be interleaved, as the determination of contexts for level information depends on the previously coded neighbor coefficients.

[0210] FIG. 24 is a flowchart illustrating another example process for significance map and coefficients level scanning and CABAC context derivation. The method of FIG. 24 slightly differs from that shown in FIG. 22, as contexts for the sub-sets are determined based on the presence of the DC coefficient in the sub-sets. As one example, the video coder may determine different sets of contexts for different sub-sets of the coefficients based on whether the respective sub-sets contain a DC coefficient of the transform coefficients (125). Like FIG. 22, steps 122, 125 and 126 may be interleaved, as the determination of contexts for level information depends on the previously coded neighbor coefficients.

[0211] FIG. 25 is a flowchart illustrating another example process for significance map and coefficients level scanning and CABAC context derivation. The method of FIG. 25 slightly differs from that shown in FIG. 22, as contexts are determined based on a weighted number of significant coefficients in other preceding subsets. As one example, the video coder may determine different sets of contexts for different sub-sets of the coefficients based a number of significant coefficients in an immediately preceding sub-set of the coefficients and a weighted number of significant coefficients in other preceding sub-sets of the coefficients (127). Like FIG. 22, steps 122, 127 and

126 may be interleaved, as the determination of contexts for level information depends on the previously coded neighbor coefficients.

[0212] FIG. 26 is a flowchart illustrating an example process for significance map coding using an inverse scan direction. A method of coding transform coefficients associated with residual video data in a video coding process is proposed. The method may be performed by a video coder, such as video encoder 20 or video decoder 30 of FIG. 4. The video coder may be configured to select a scan order with an inverse direction (140) and to determine contexts for context adaptive binary arithmetic coding (CABAC) of the information indicating a current one of the significant coefficients based on previously coded significant coefficients in the inverse scan direction (142). The video coder may further be configured code information indicating the significant transform coefficients along the inverse scan direction to form a significance map (146).

[0213] In one example, the scan has a diagonal pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides. In another example, the scan has a horizontal pattern and the previously coded significant coefficients reside at positions below a scan line on which a current one of the significant coefficients resides. In another example, the scan has a vertical pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

[0214] The video coder may further be configured to code information indicating levels of the significant transform coefficients (148). The step of coding information indicating levels of the significant transform coefficients may proceed in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients. Like FIG. 22, steps 142, 146 and 148 may be interleaved, as the determination of contexts for level information depends on the previously coded neighbor coefficients.

[0215] FIG. 27 is a flowchart illustrating an example process for significance map and coefficient level scanning according to sub-sets of transform coefficients. A method of coding transform coefficients associated with residual video data in a video coding process is proposed. The method may be performed by a video coder, such as video encoder 20 or video decoder 30 of FIG. 4. The video coder may be configured to arrange a block of transform coefficients into one or more sub-sets of transform coefficients (160), code significance of transform coefficients in each sub-set (162), and

code levels of transform coefficients in each sub-set (164). In one example, arranging the block of transform coefficients may include arranging the block of transform coefficients into a single set of transform coefficients corresponding to an entire transform unit. In another example, arranging the block of transform coefficients may include arranging the block of transform coefficients into one or more sub-sets of transform coefficients based on a scan order.

[0216] The video coder may be configured to code the significance of transform coefficients in each sub-set according to a scan order, and code the levels of transform coefficients according to the scan order. Coding the significance map (162) and the levels (164) may be performed together in two or more consecutive scan passes on the sub-set (165).

[0217] FIG. 28 is a flowchart illustrating another example process for significance map and coefficient level scanning according to sub-sets of transform coefficients. The video encoder may perform the consecutive scans (165) by first coding the significance of transform coefficients in a sub-set in a first scan of transform coefficients in the respective sub-set (170).

[0218] Coding of the coefficient levels (164) in each sub-set includes at least a second scan of transform coefficients in the respective sub-set. The second scan may include coding bin one of the levels of transform coefficients in a sub-set in a second scan of transform coefficients in the respective sub-set (172), coding remaining bins of the levels of transform coefficients in a sub-set in a third scan of transform coefficients in the respective sub-set (174), and coding a sign of the levels of transform coefficients in a sub-set in a fourth scan of transform coefficients in the respective sub-set (176).

[0219] FIG. 29 is a flowchart illustrating another example process for significance map and coefficient level scanning according to sub-sets of transform coefficients. In this example, coding of the sign of the levels of the transform coefficients (176) is performed before the coding the levels (172, 174).

[0220] FIG. 30 is a flowchart illustrating an example process for entropy coding using multiple regions. A method of coding a plurality of transform coefficients associated with residual video data in a video coding process is proposed. The method may be performed by a video coder, such as video encoder 20 or video decoder 30 of FIG. 4. The video coder may be configured to code information indicating significant coefficients for the plurality of transform coefficients according to a scan order (180), divide the coded information into a first region and a second region (182), entropy code

the coded information in the first region according to a first set of contexts using context adaptive binary arithmetic coding (184), and entropy code the coded information in the second region according to a second set of contexts using context adaptive binary arithmetic coding (186). In one example, the scan order has an inverse direction and a diagonal scan pattern. This method may also be applied to more than two regions, wherein each region has a set of contexts.

[0221] The first and second regions may be divided in several ways. In one example, the first region contains at least the DC component of the plurality of transform coefficients, and the second region contains the remaining plurality of transform coefficients not in the first region.

[0222] In another example, the first region contains all transform coefficients within a region defined by $x+y<T$, where x is the horizontal position of the transform coefficient, y is the vertical position of the transform coefficient, and T is a threshold. The first region may contain the DC coefficient. The second region contains the remaining plurality of transform coefficients not in the first region.

[0223] In another example, the first region contains all transform coefficients within a region defined by $x<T$ and $y<T$, where x is the horizontal position of the transform coefficient, y is the vertical position of the transform coefficient, and T is a threshold. The second region contains the remaining plurality of transform coefficients not in the first region.

[0224] In another example, the first region contains the DC coefficient, the second region contains all transform coefficients (excluding the DC coefficient) within a region defined by $x<T$ and $y<T$, where x is the horizontal position of the transform coefficient, y is the vertical position of the transform coefficient, and T is a threshold, and the third regions contains the remaining plurality of transform coefficients not in the first region or the second region. In another example, the second and third regions described above may use the same method to derive contexts, but different context sets for each region are used.

[0225] In another example, the first region includes the DC component and the transform coefficients at positions (1,0) and (0,1). The second region contains the remaining plurality of transform coefficients not in the first region.

[0226] In another example, the first region contains only the DC component of the plurality of transform coefficients, and the second region contains the remaining plurality of transform coefficients.

[0227] In general, the first context for each transform coefficient in the first region is based on the position of each transform coefficient in the first region, while the second context for each transform coefficient in the second region is based on the coded information of causal neighbors of each transform coefficient. In some examples, the second context is further based on the position of each transform coefficient in the second region. In another example, the second context for each transform coefficient in the second region is based on the coded information of five causal neighbors of each transform coefficient.

[0228] In one or more examples, the functions described in this disclosure may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be performed by a hardware-based processing unit, such as one or more processors, that execute the software in the form of computer-readable instructions or code. Such instructions or code may be stored on or transmitted over a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible, non-transitory medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0229] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, Flash memory, CD-ROM or any other solid state, optical or magnetic data storage media, including optical disk storage, magnetic disk storage, or other magnetic storage devices, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the

coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that tangible computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transient media, but are instead directed to non-transient, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc, where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0230] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0231] The techniques of this disclosure may be performed by a wide variety of devices or apparatuses, including desktop computers, notebook (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called smartphones, televisions, cameras, display devices, digital media players, video gaming consoles, or the like. In many cases, such devices may be equipped for wireless communication. In addition, such techniques may be implemented by an integrated circuit (IC) or a set of ICs (e.g., a chip set). A device configured to perform the techniques of the this disclosure may include any of the devices mentioned above and, in some cases, may be a video encoder or video decoder, or combined video encoder-decoder, i.e., a video CODEC, which may be formed by a combination of hardware, software, and firmware. Various components, modules, or units may be described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative

hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0232] Various examples have been described. These and other examples are within the scope of the following claims.

CLAIMS:

1. A method of coding transform coefficients associated with residual video data in a video coding process, the method comprising:

coding a respective syntax element for each transform coefficient in a block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and

coding information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

2. The method of claim 1, further comprising:

determining contexts for context adaptive binary arithmetic coding (CABAC) of each respective syntax element indicating a current one of the significant coefficients based on previously coded significant coefficients in the inverse scan direction.

3. The method of claim 2, wherein, if a previously coded significant coefficient is located outside of a transform block, coded level information for the previously coded significant coefficient is assumed to be zero.

4. The method of claim 2, wherein the first scan pass has a diagonal pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

5. The method of claim 2, wherein the first scan pass has a horizontal pattern and the previously coded significant coefficients reside at positions below a scan line on which a current one of the significant coefficients resides.

6. The method of claim 2, wherein the first scan pass has a vertical pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

7. A system for coding transform coefficients associated with residual video data in a video coding process, the system comprising:

a memory configured to store a block of transform coefficients; and

a video coding processor comprising an integrated circuit, the video coding processor configured to:

code a respective syntax element for each transform coefficient in the block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and

code information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

8. The system of claim 7, wherein the video coding processor is further configured to determine contexts for context adaptive binary arithmetic coding (CABAC) of each respective syntax element indicating a current one of the significant coefficients based on previously coded significant coefficients in the inverse scan direction.

9. The system of claim 8, wherein, if a previously coded significant coefficient is located outside of a transform block, coded level information for the previously coded significant coefficient is assumed to be zero.

10. The system of claim 8, wherein the first scan pass has a diagonal pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

11. The system of claim 8, wherein the first scan pass has a horizontal pattern and the previously coded significant coefficients reside at positions below a scan line on which a current one of the significant coefficients resides.

12. The system of claim 8, wherein the first scan pass has a vertical pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

13. A system for coding transform coefficients associated with residual video data in a video coding process, the system comprising:

means for coding a respective syntax element for each transform coefficient in a block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and

means for coding information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

14. The system of claim 13, further comprising:

means for determining contexts for context adaptive binary arithmetic coding (CABAC) of each respective syntax element indicating a current one of the significant coefficients based on previously coded significant coefficients in the inverse scan direction.

15. The system of claim 14, wherein the first scan pass has a diagonal pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

16. The system of claim 14, wherein the first scan pass has a horizontal pattern and the previously coded significant coefficients reside at positions below a scan line on which a current one of the significant coefficients resides.

17. The system of claim 14, wherein the first scan pass has a vertical pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

18. A non-transitory computer-readable storage medium having stored thereon instructions that, when executed, cause one or more processors of a device for coding transform coefficients associated with residual video data in a video coding process to:

code a respective syntax element for each transform coefficient in a block of transform coefficients with a first scan pass, wherein each respective syntax element indicates whether or not a corresponding transform coefficient in the block of transform coefficients is a significant coefficient having an absolute value greater than zero, and wherein the first scan pass proceeds in an inverse scan direction from higher frequency coefficients in the block of transform coefficients to lower frequency coefficients in the block of transform coefficients; and

coding information indicating levels of the significant transform coefficients in the block of transform coefficients with a second scan pass, wherein the second scan pass proceeds in the inverse scan direction.

19. The non-transitory computer-readable storage medium of claim 18, further causing the one or more processors to:

determine contexts for context adaptive binary arithmetic coding (CABAC) of each respective syntax element indicating a current one of the significant coefficients based on previously coded significant coefficients in the inverse scan direction.

20. The non-transitory computer-readable storage medium of claim 19, wherein the first scan pass has a diagonal pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

21. The non-transitory computer-readable storage medium of claim 19, wherein the first scan pass has a horizontal pattern and the previously coded significant coefficients reside at positions below a scan line on which a current one of the significant coefficients resides.

22. The non-transitory computer-readable storage medium of claim 19, wherein the first scan pass has a vertical pattern and the previously coded significant coefficients reside at positions to the right of a scan line on which a current one of the significant coefficients resides.

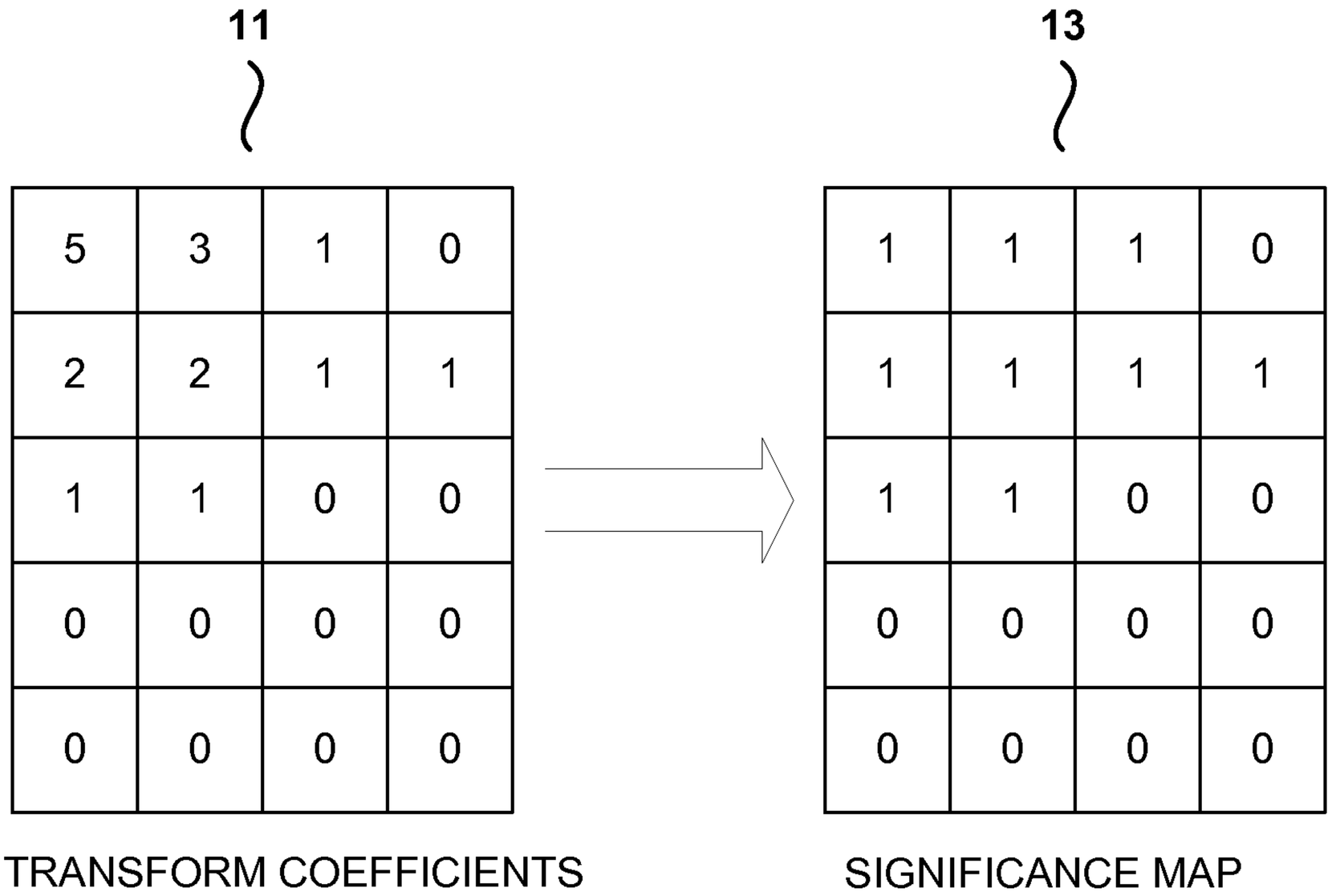

11
13
5 3 1 0
2 2 1 1
1 1 0 0
0 0 0 0
0 0 0 0
1 1 1 0
1 1 1 1
1 1 0 0
0 0 0 0
0 0 0 0
TRANSFORM COEFFICIENTS
SIGNIFICANCE MAP

FIG. 1

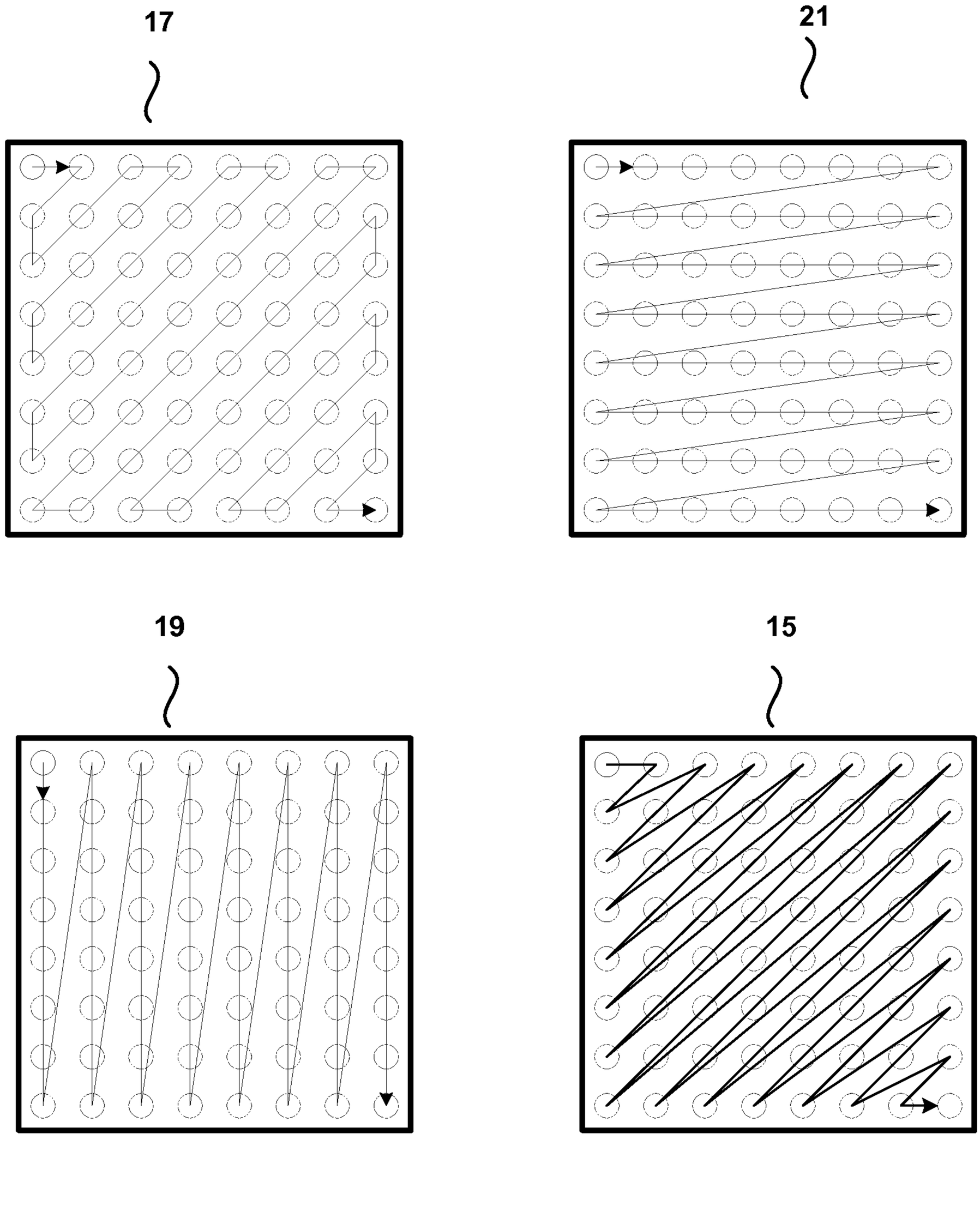


FIG. 2

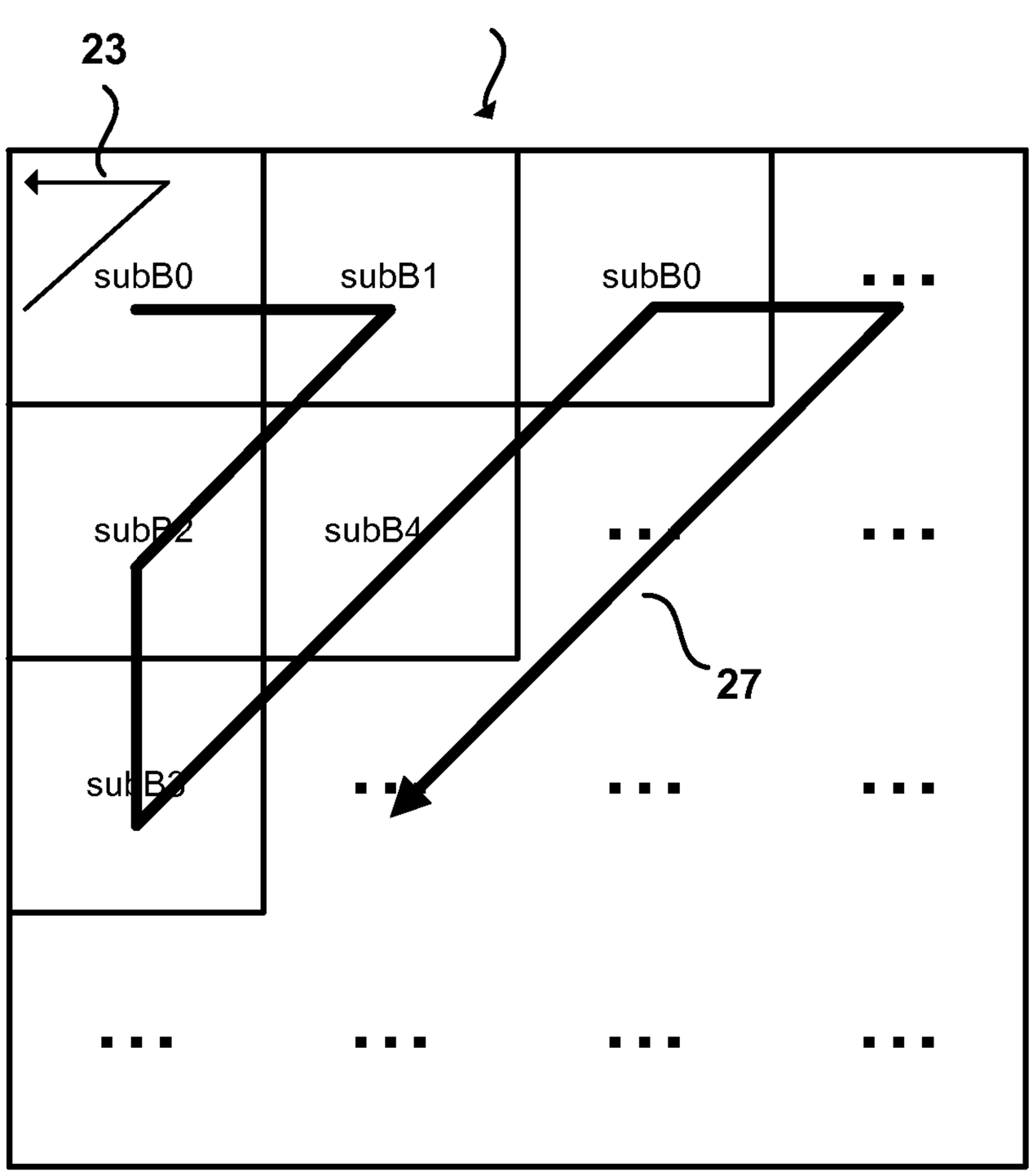


FIG. 3

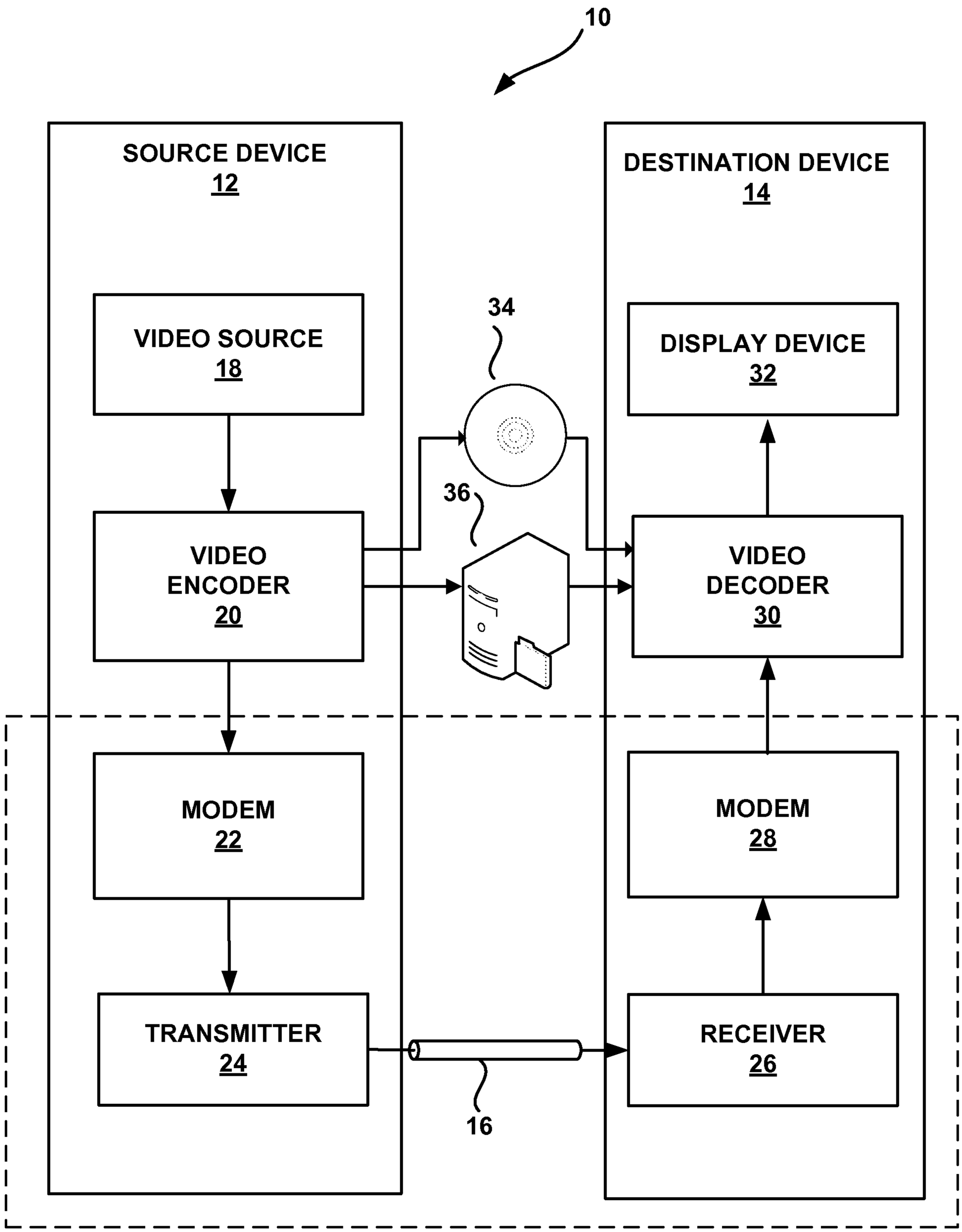

10
SOURCE DEVICE
12
DESTINATION DEVICE
14
VIDEO SOURCE
18
34
DISPLAY DEVICE
32
36
VIDEO
ENCODER
20
VIDEO
DECODER
30
MODEM
22
MODEM
28
TRANSMITTER
24
RECEIVER
26
16

FIG. 4

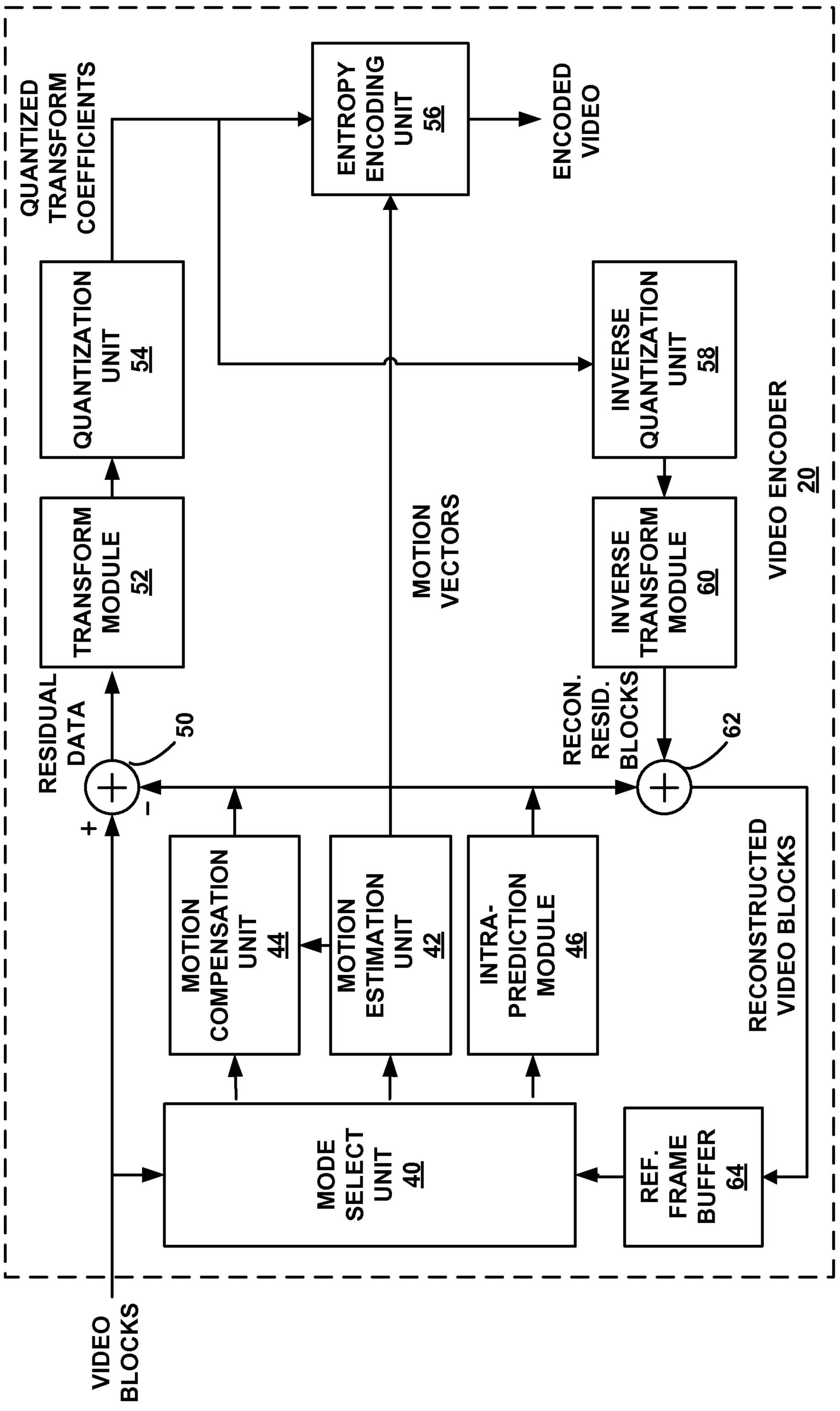

VIDEO ENCODER
20
VIDEO BLOCKS
RESIDUAL DATA
+
-
50
TRANSFORM MODULE
52
QUANTIZATION UNIT
54
QUANTIZED TRANSFORM COEFFICIENTS
ENTROPY ENCODING UNIT
56
ENCODED VIDEO
INVERSE QUANTIZATION UNIT
58
INVERSE TRANSFORM MODULE
60
RECON. RESID. BLOCKS
62
MOTION VECTORS
MOTION COMPENSATION UNIT
44
MOTION ESTIMATION UNIT
42
INTRA-PREDICTION MODULE
46
MODE SELECT UNIT
40
REF. FRAME BUFFER
64
RECONSTRUCTED VIDEO BLOCKS

FIG. 5

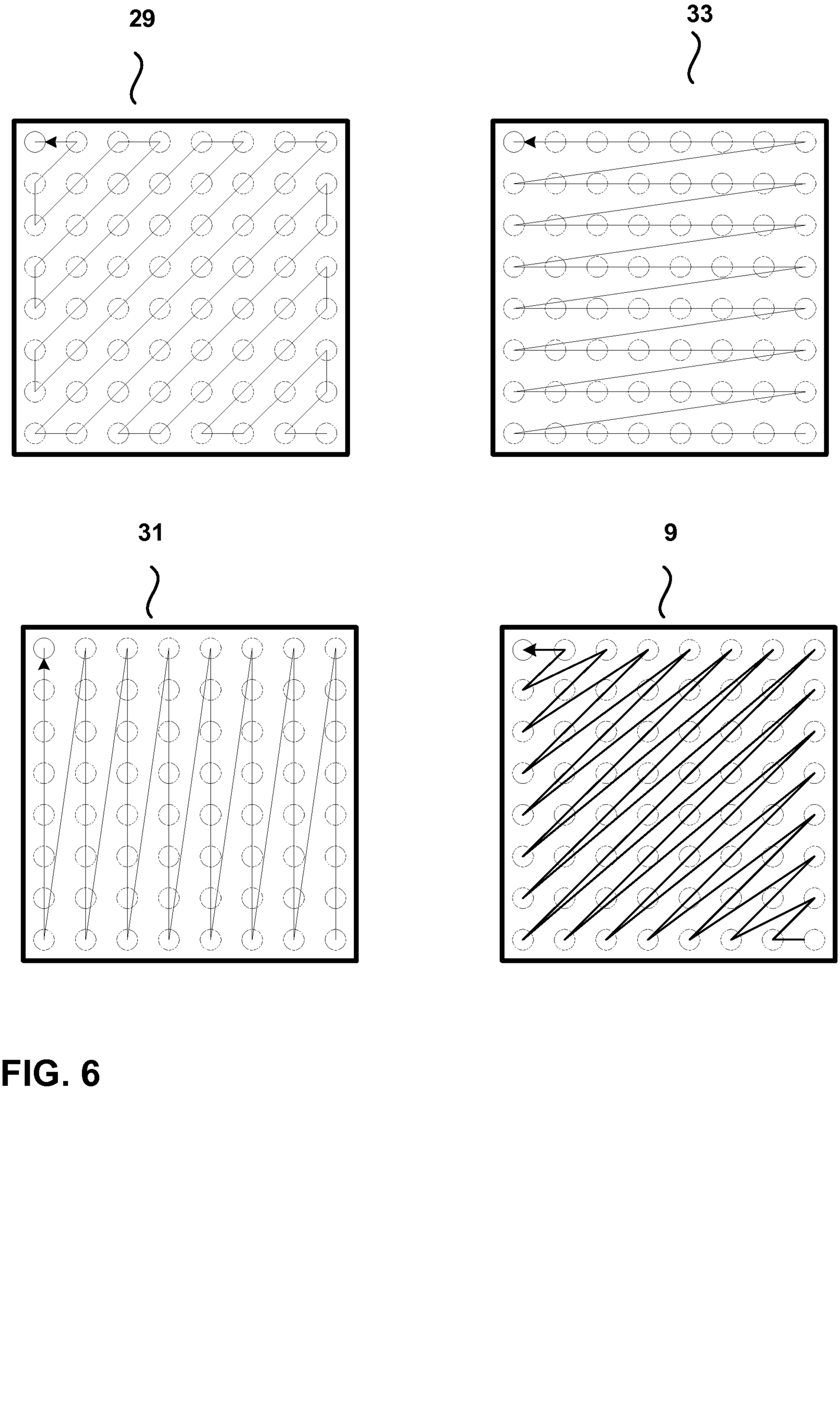


FIG. 6

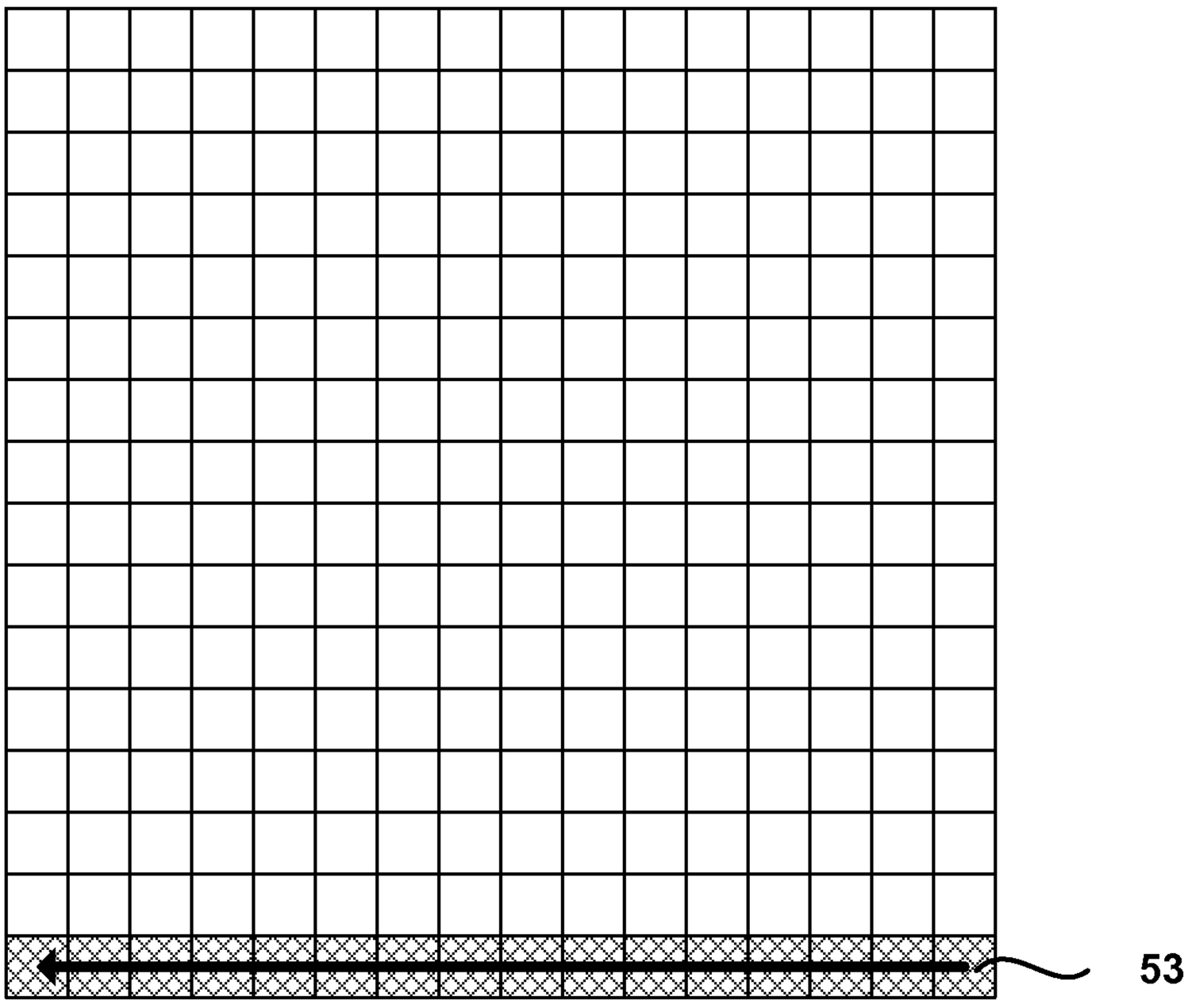


FIG. 8

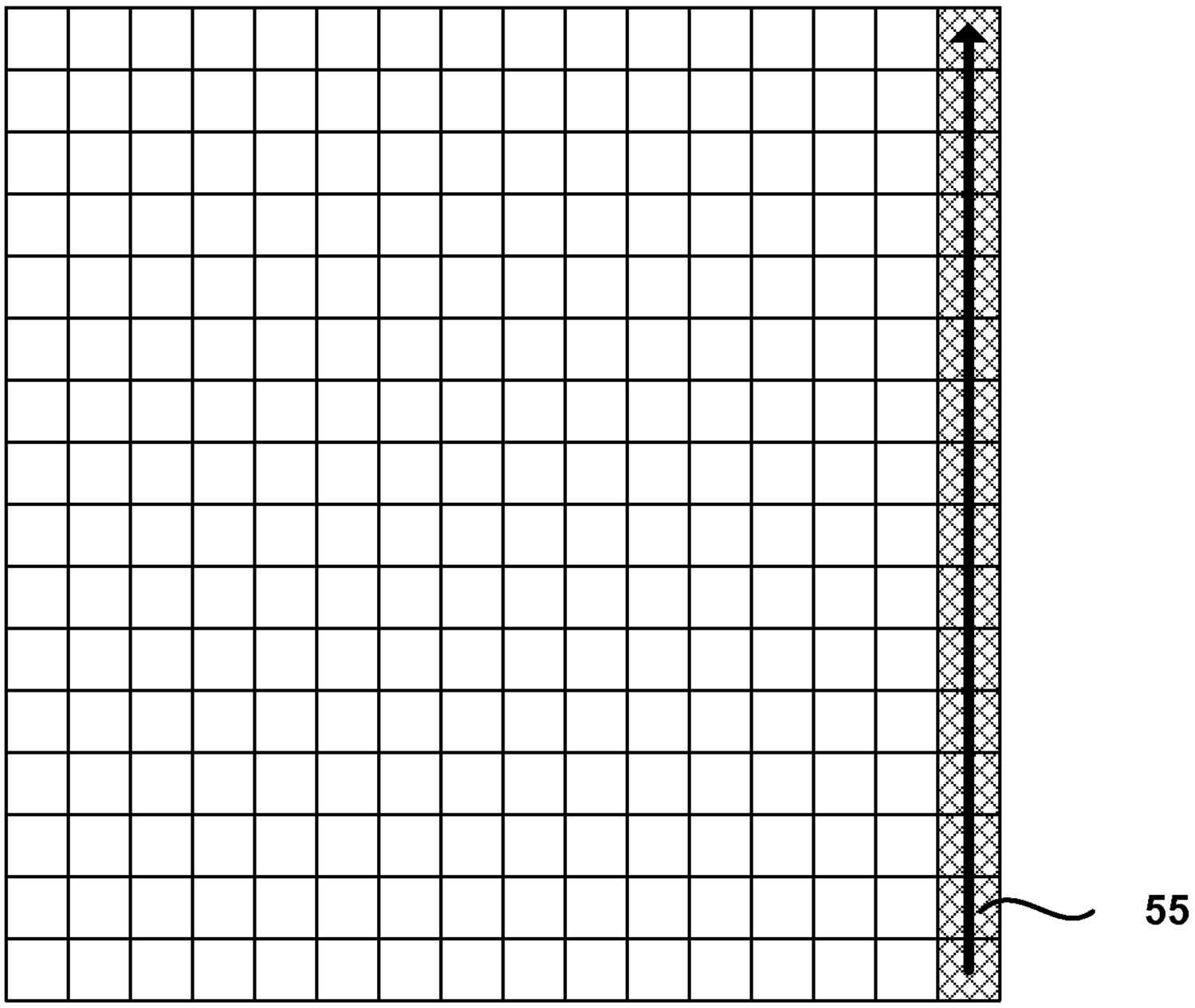


FIG. 9

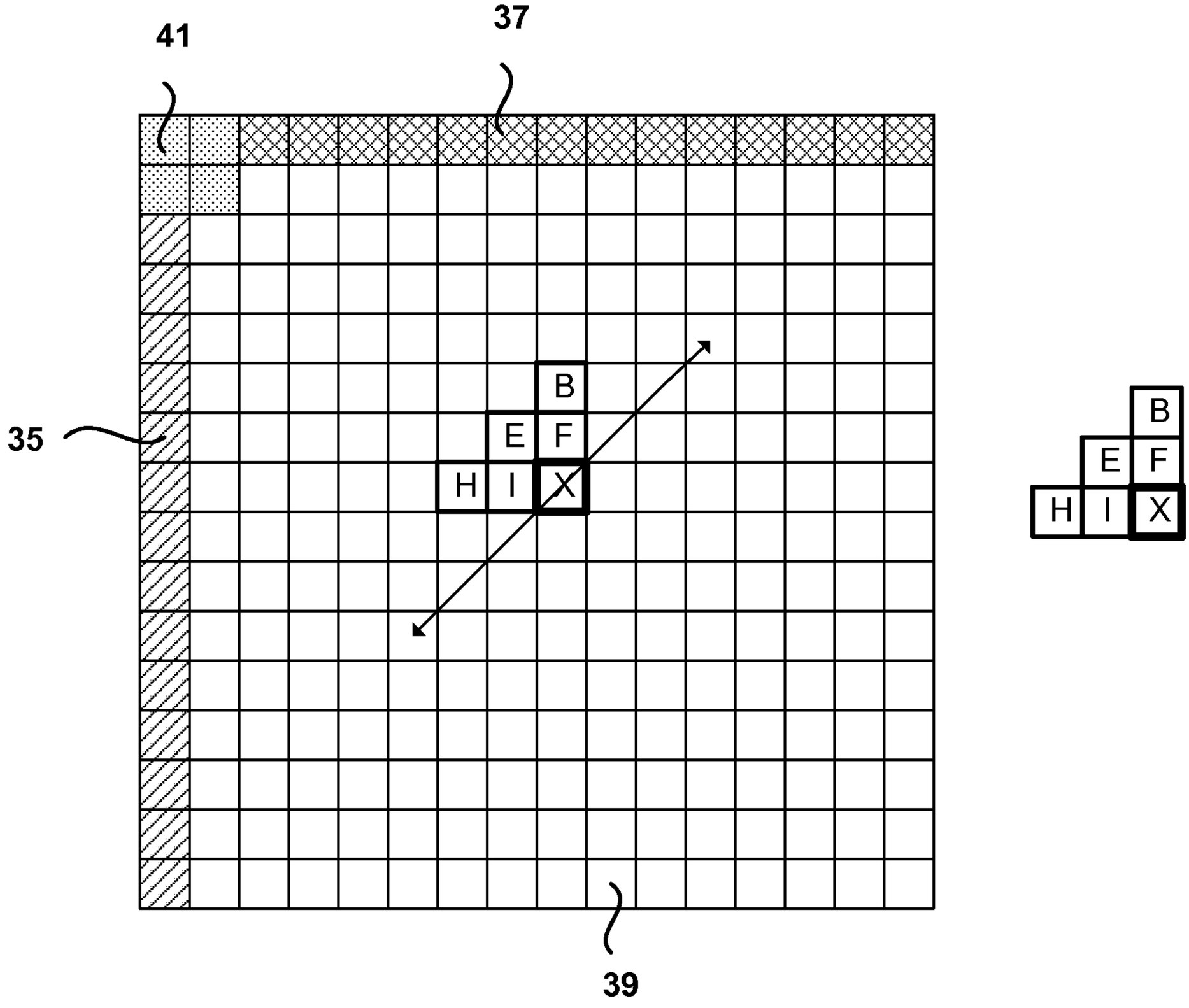


FIG. 10

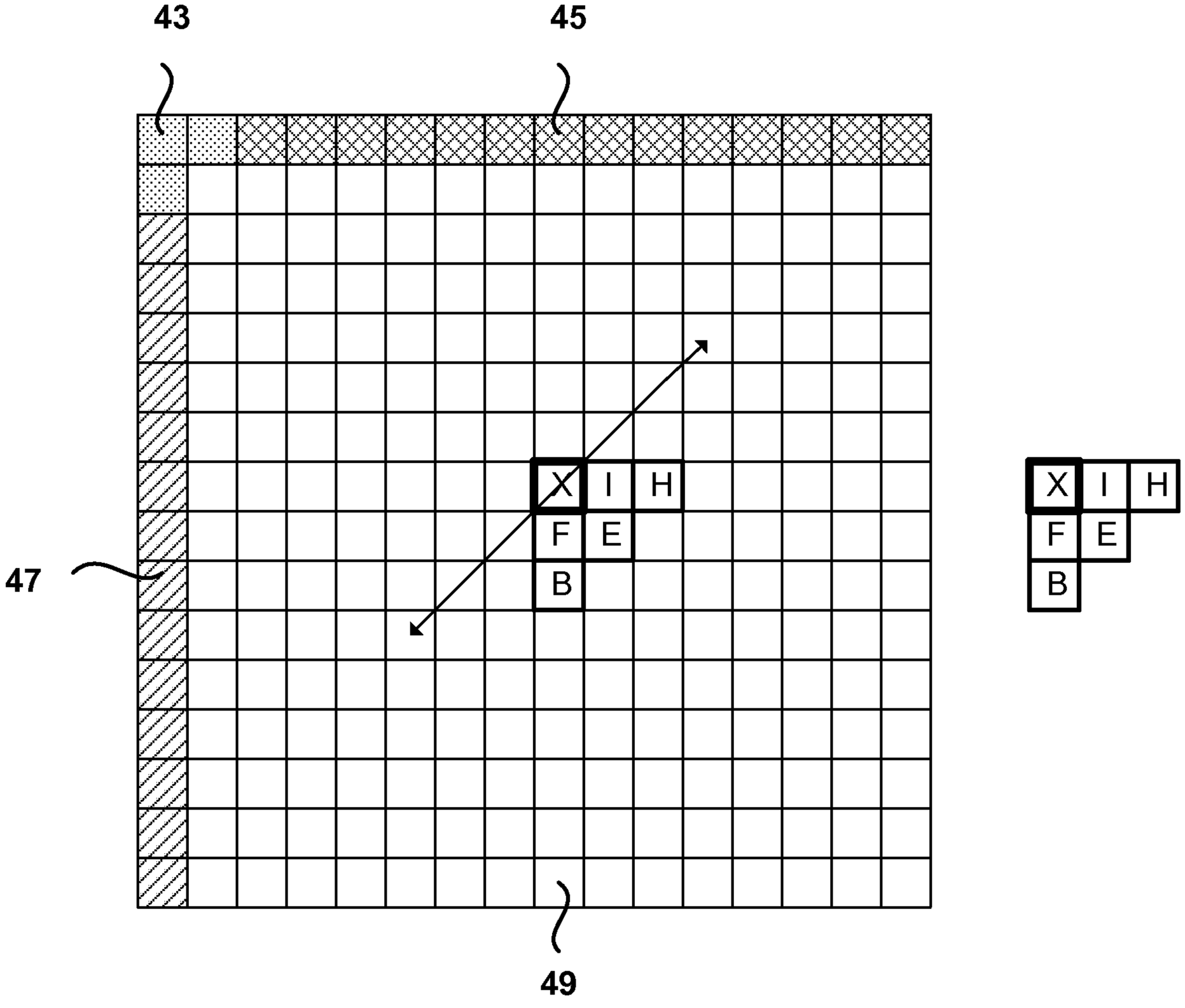


FIG. 11

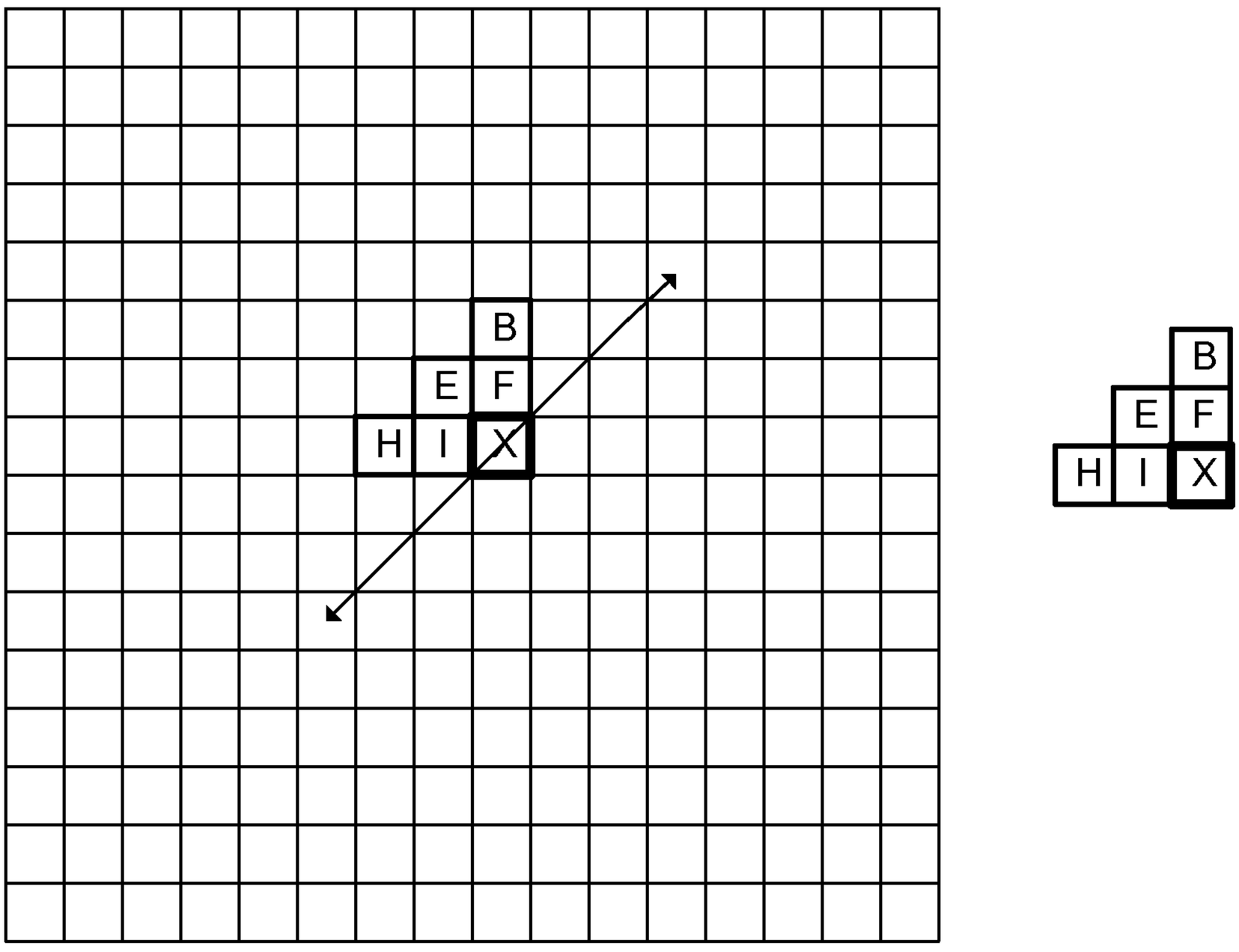


FIG. 12

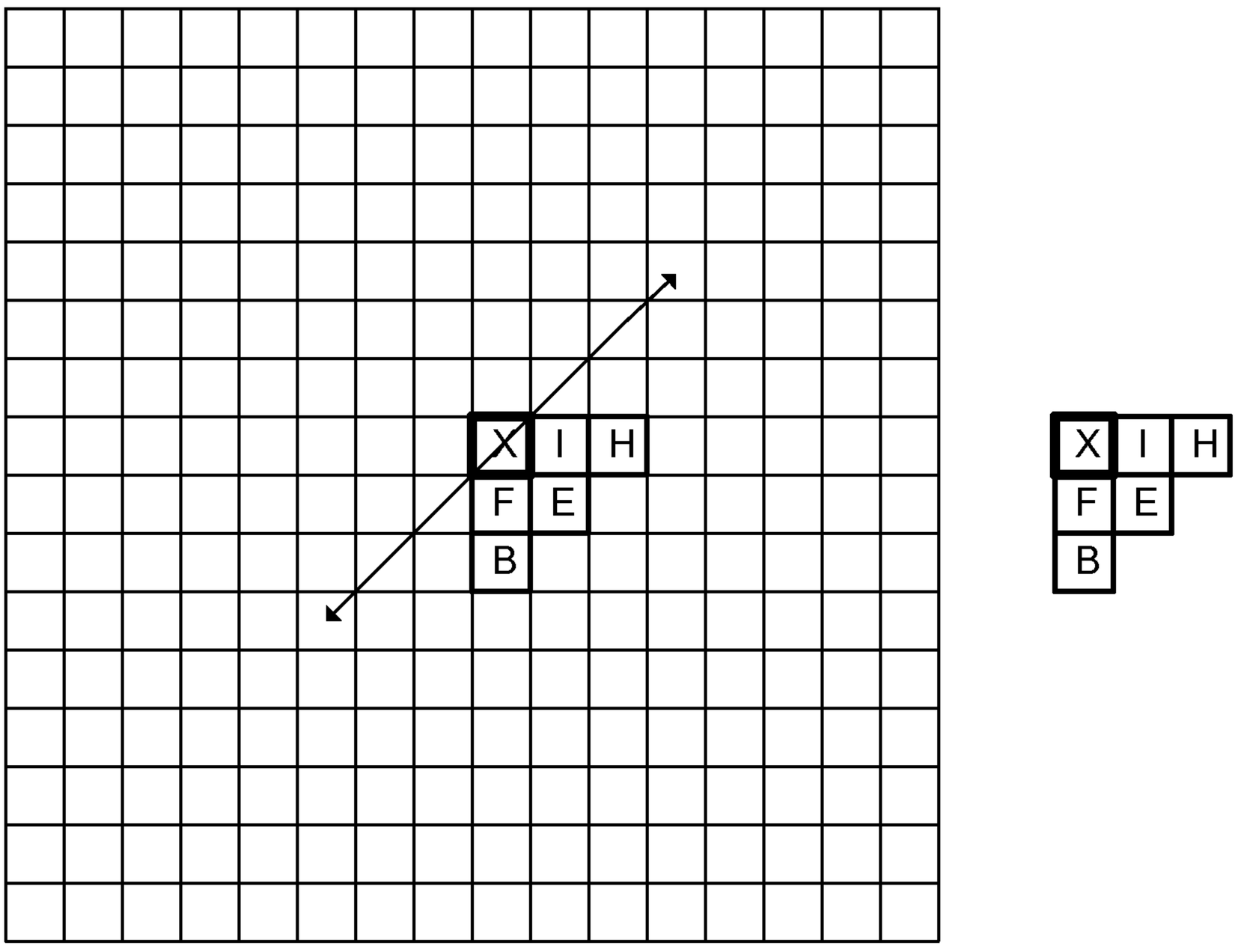


FIG. 13

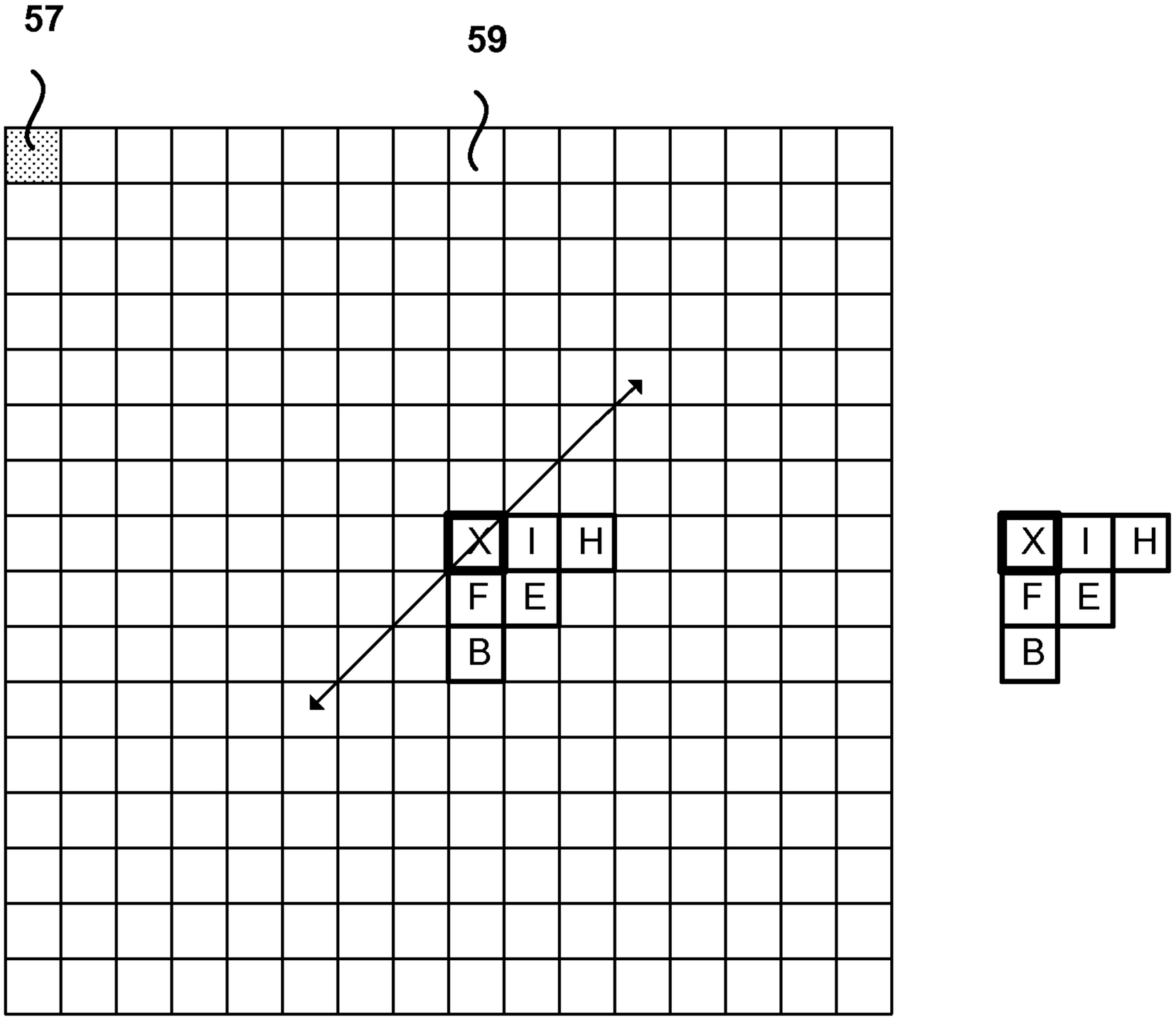


FIG. 14

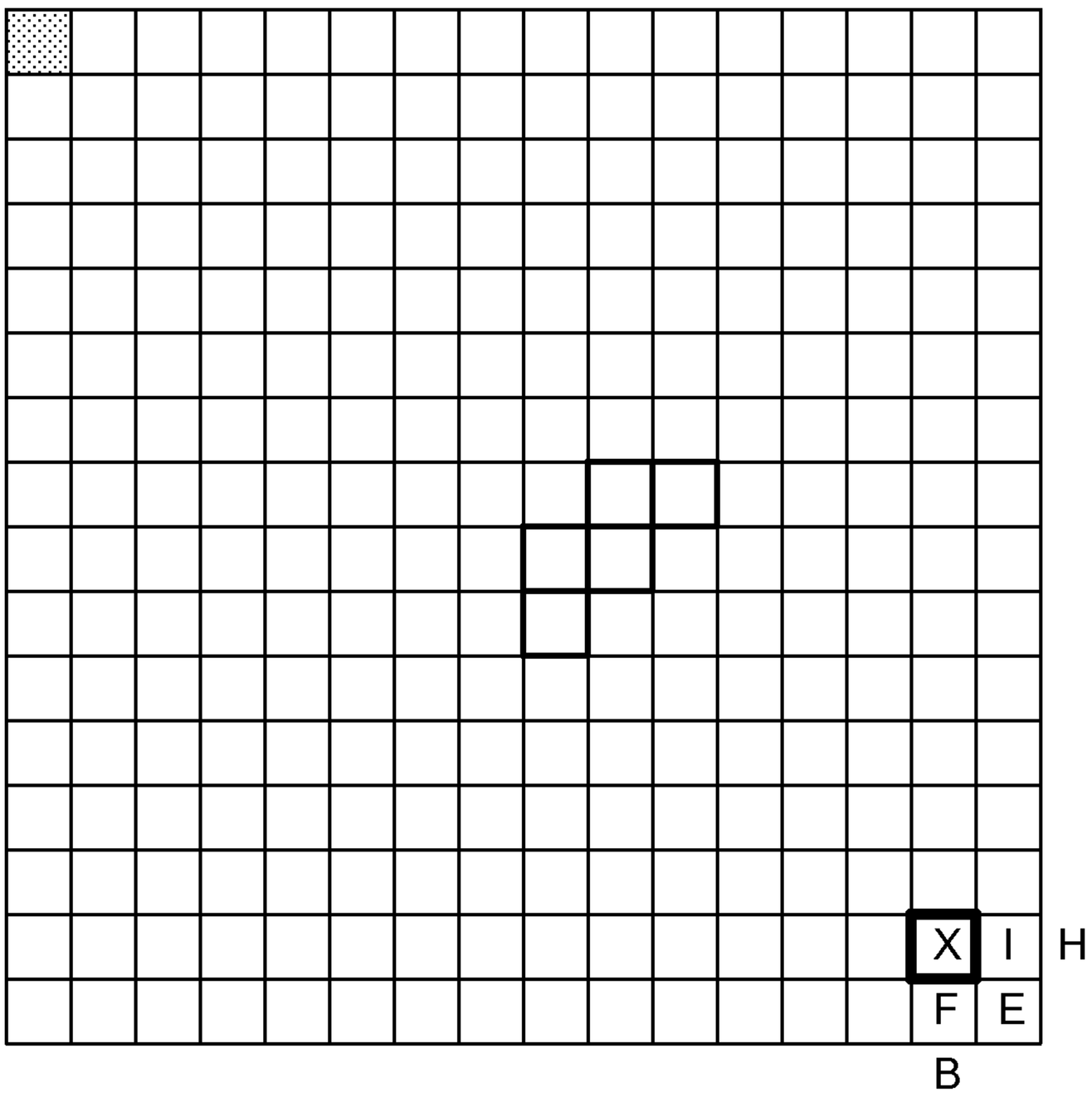


FIG. 15

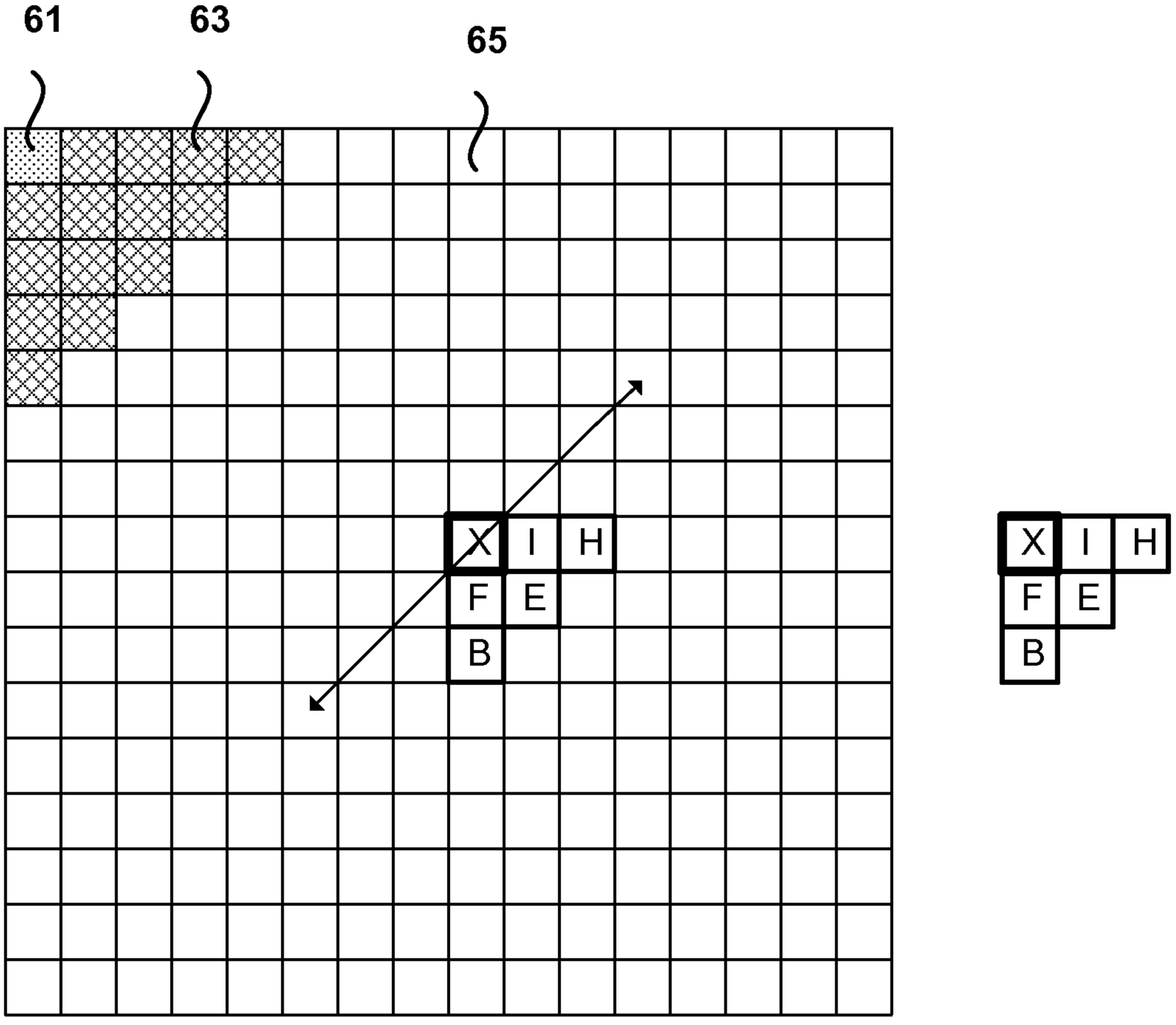


FIG. 16

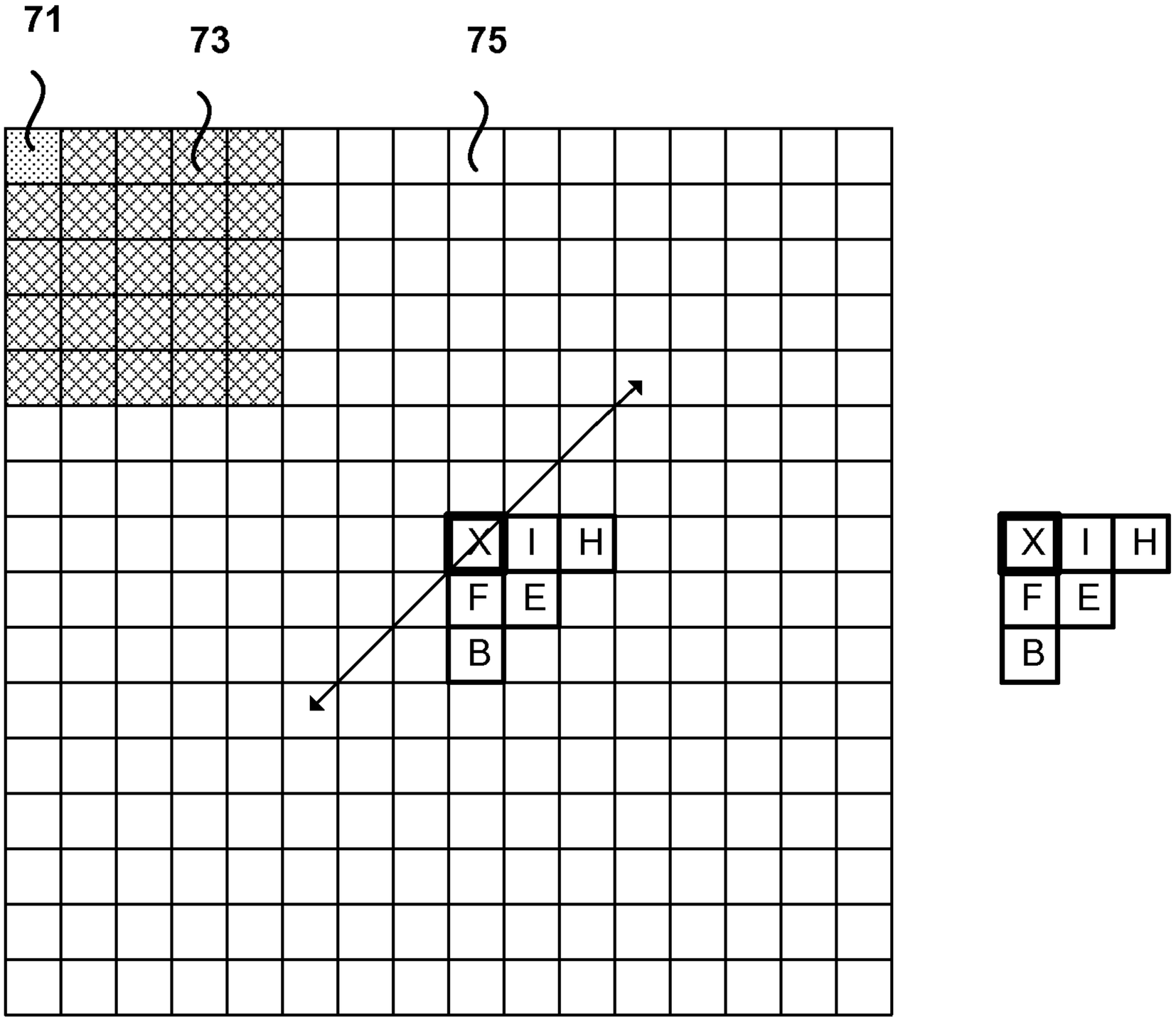


FIG. 17

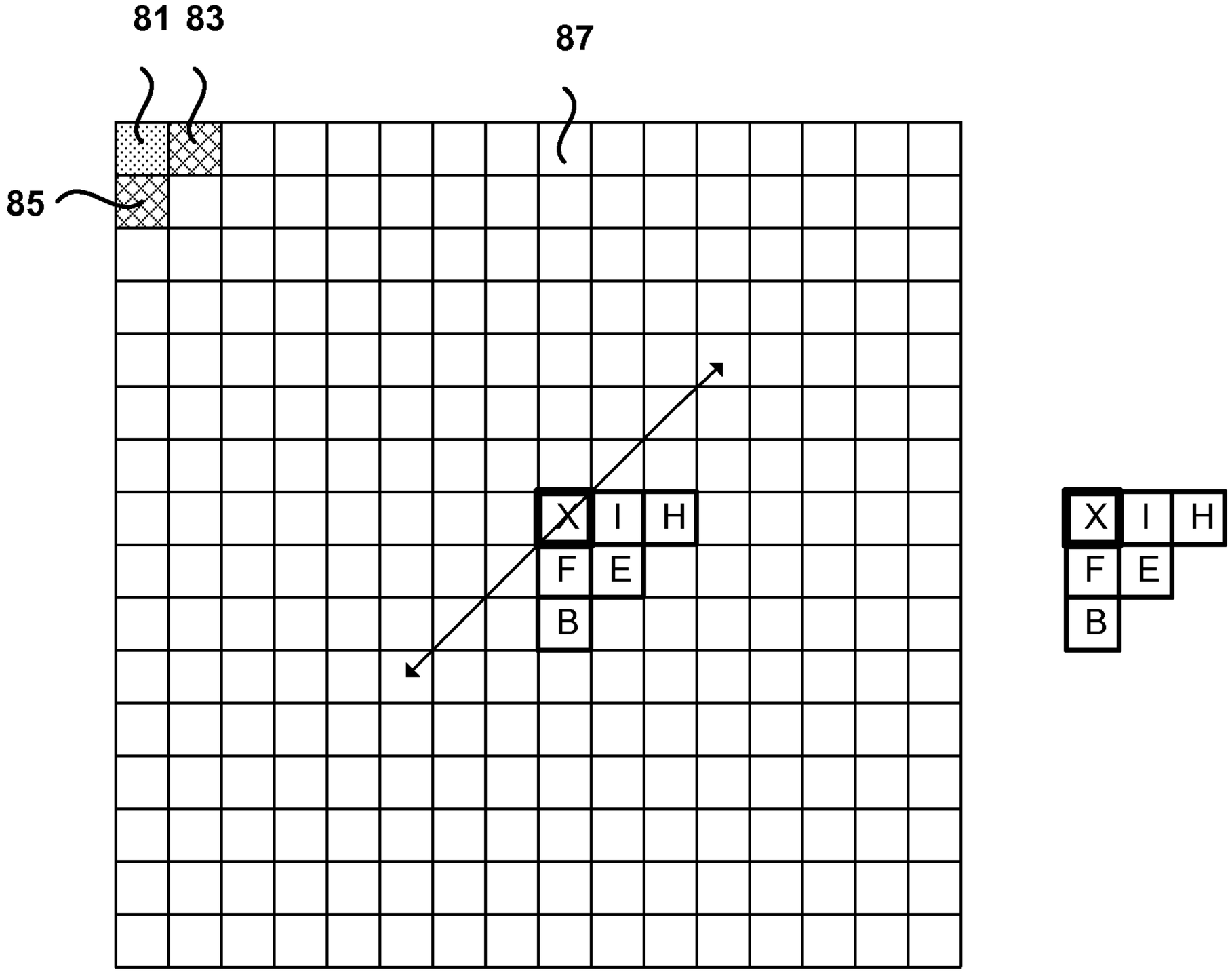


FIG. 18

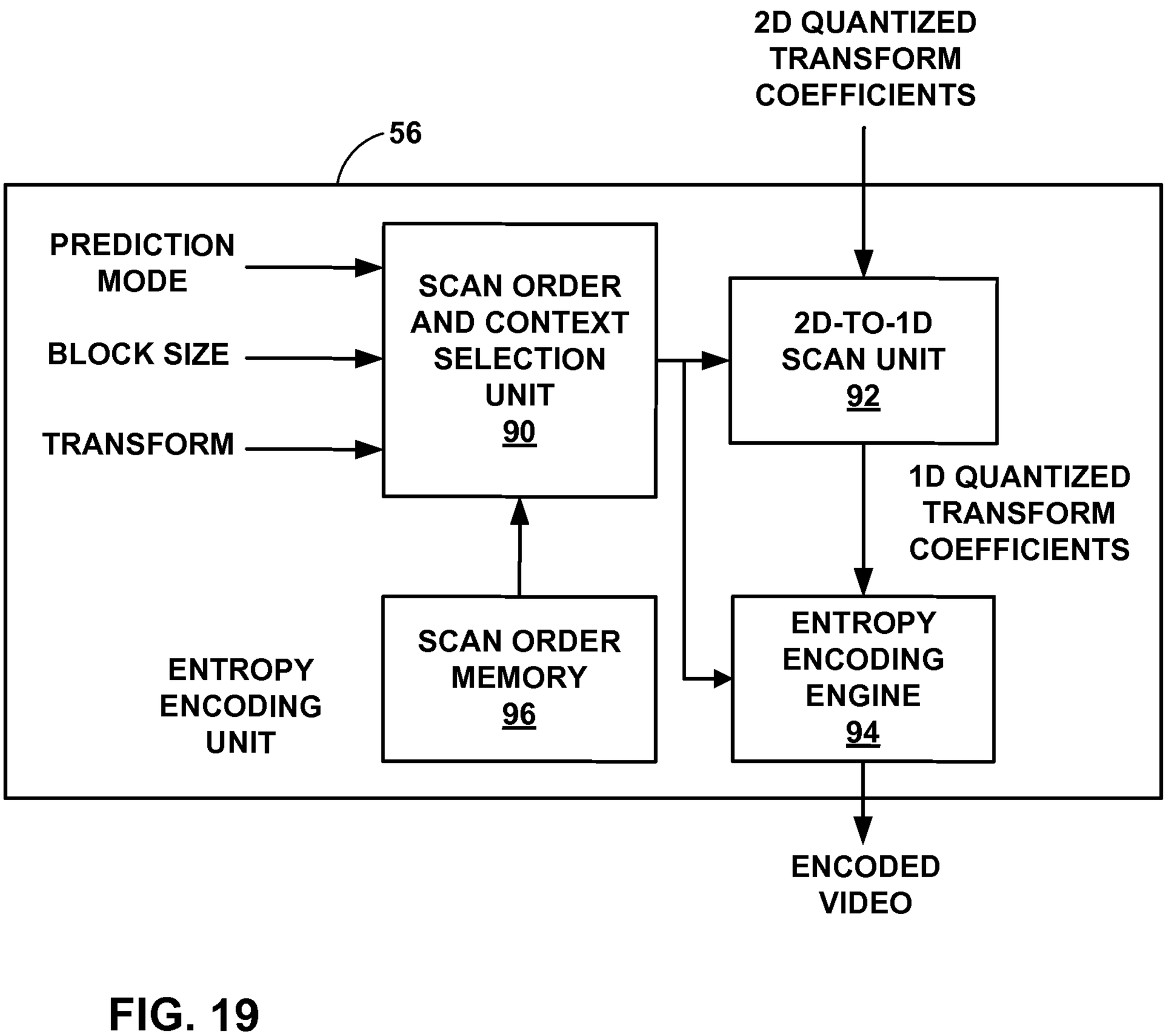

2D QUANTIZED TRANSFORM COEFFICIENTS
56
PREDICTION MODE
BLOCK SIZE
TRANSFORM
SCAN ORDER AND CONTEXT SELECTION UNIT 90
2D-TO-1D SCAN UNIT 92
1D QUANTIZED TRANSFORM COEFFICIENTS
SCAN ORDER MEMORY 96
ENTROPY ENCODING ENGINE 94
ENTROPY ENCODING UNIT
ENCODED VIDEO

FIG. 19

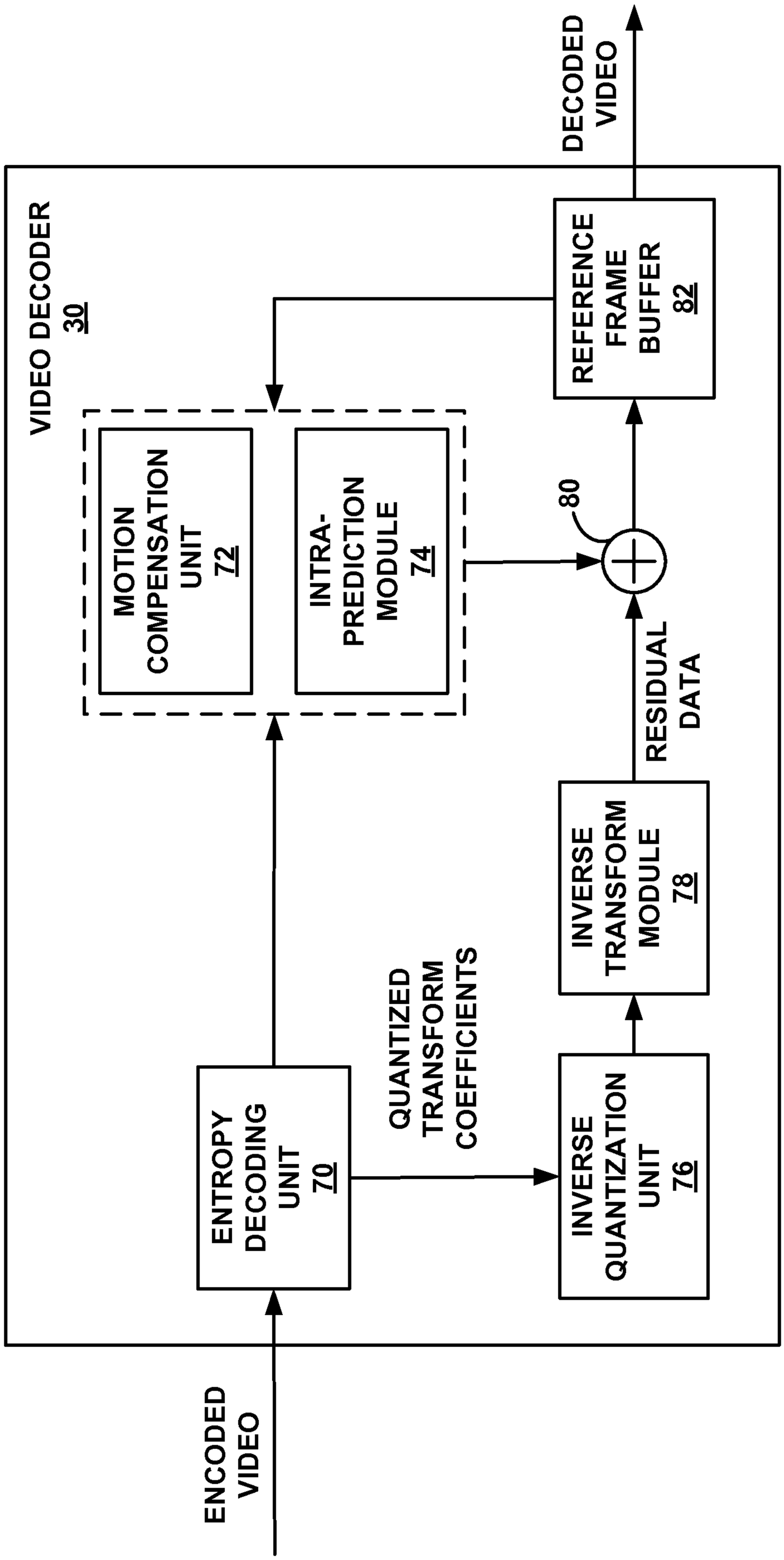

DECODED VIDEO
VIDEO DECODER 30
REFERENCE FRAME BUFFER 82
MOTION COMPENSATION UNIT 72
INTRA-PREDICTION MODULE 74
80
RESIDUAL DATA
INVERSE TRANSFORM MODULE 78
INVERSE QUANTIZATION UNIT 76
QUANTIZED TRANSFORM COEFFICIENTS
ENTROPY DECODING UNIT 70
ENCODED VIDEO

FIG. 20

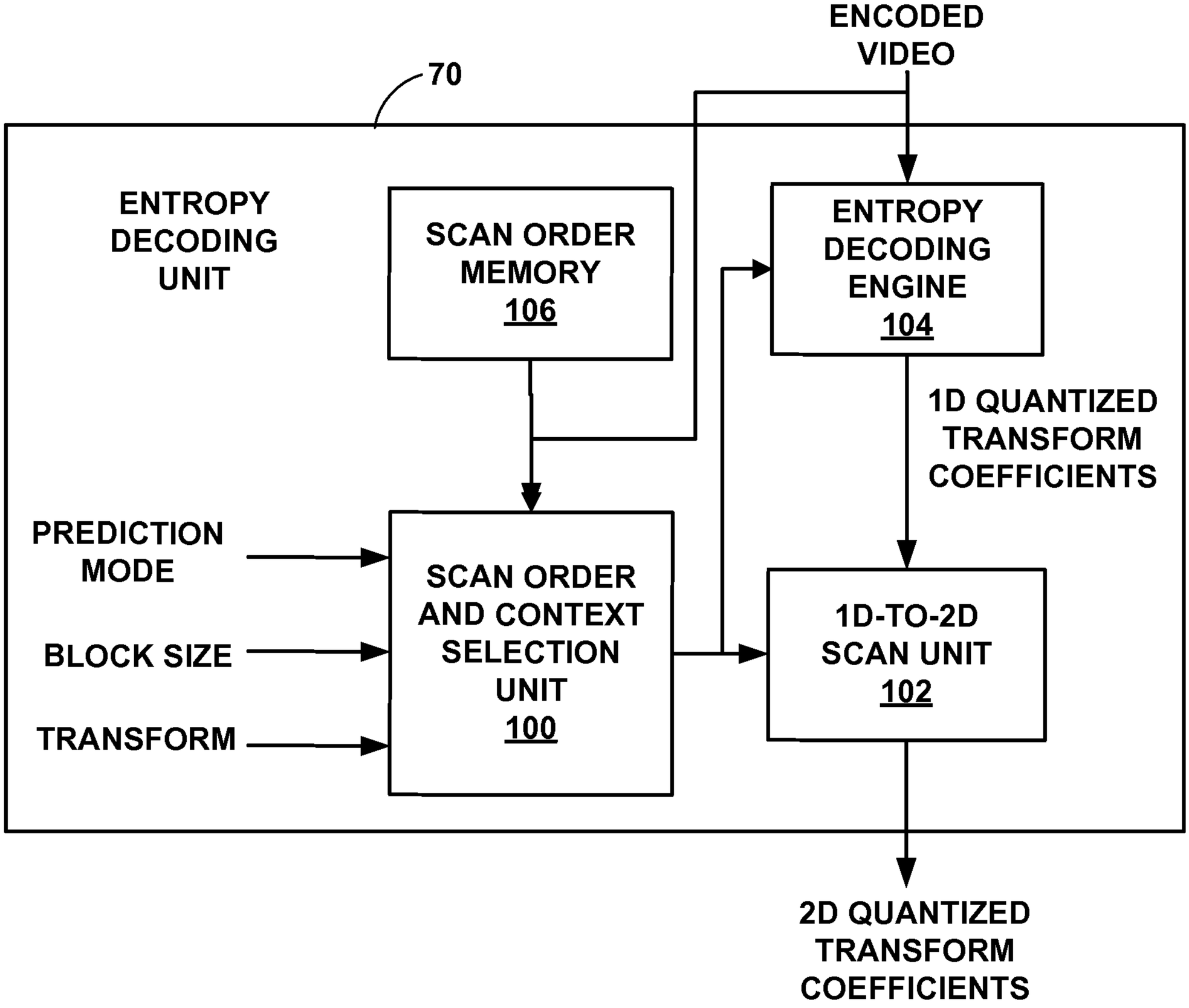

70
ENTROPY DECODING UNIT
SCAN ORDER MEMORY 106
ENTROPY DECODING ENGINE 104
VIDEO
ENCODED
1D QUANTIZED TRANSFORM COEFFICIENTS
PREDICTION MODE
BLOCK SIZE
TRANSFORM
SCAN ORDER AND CONTEXT SELECTION UNIT 100
1D-TO-2D SCAN UNIT 102
2D QUANTIZED TRANSFORM COEFFICIENTS

FIG. 21

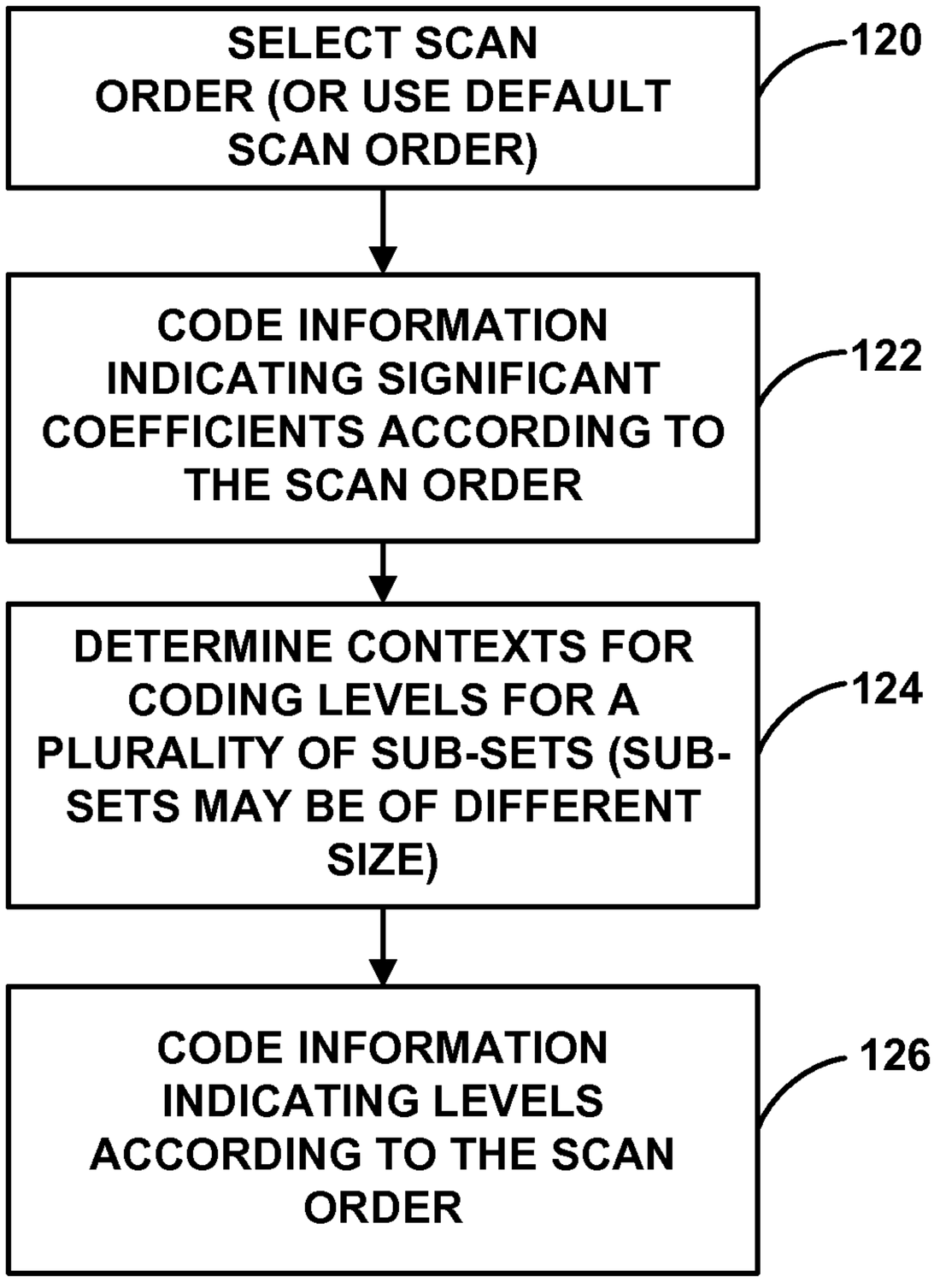

SELECT SCAN ORDER (OR USE DEFAULT SCAN ORDER)
120
CODE INFORMATION INDICATING SIGNIFICANT COEFFICIENTS ACCORDING TO THE SCAN ORDER
122
DETERMINE CONTEXTS FOR CODING LEVELS FOR A PLURALITY OF SUB-SETS (SUB-SETS MAY BE OF DIFFERENT SIZE)
124
CODE INFORMATION INDICATING LEVELS ACCORDING TO THE SCAN ORDER
126

FIG. 22

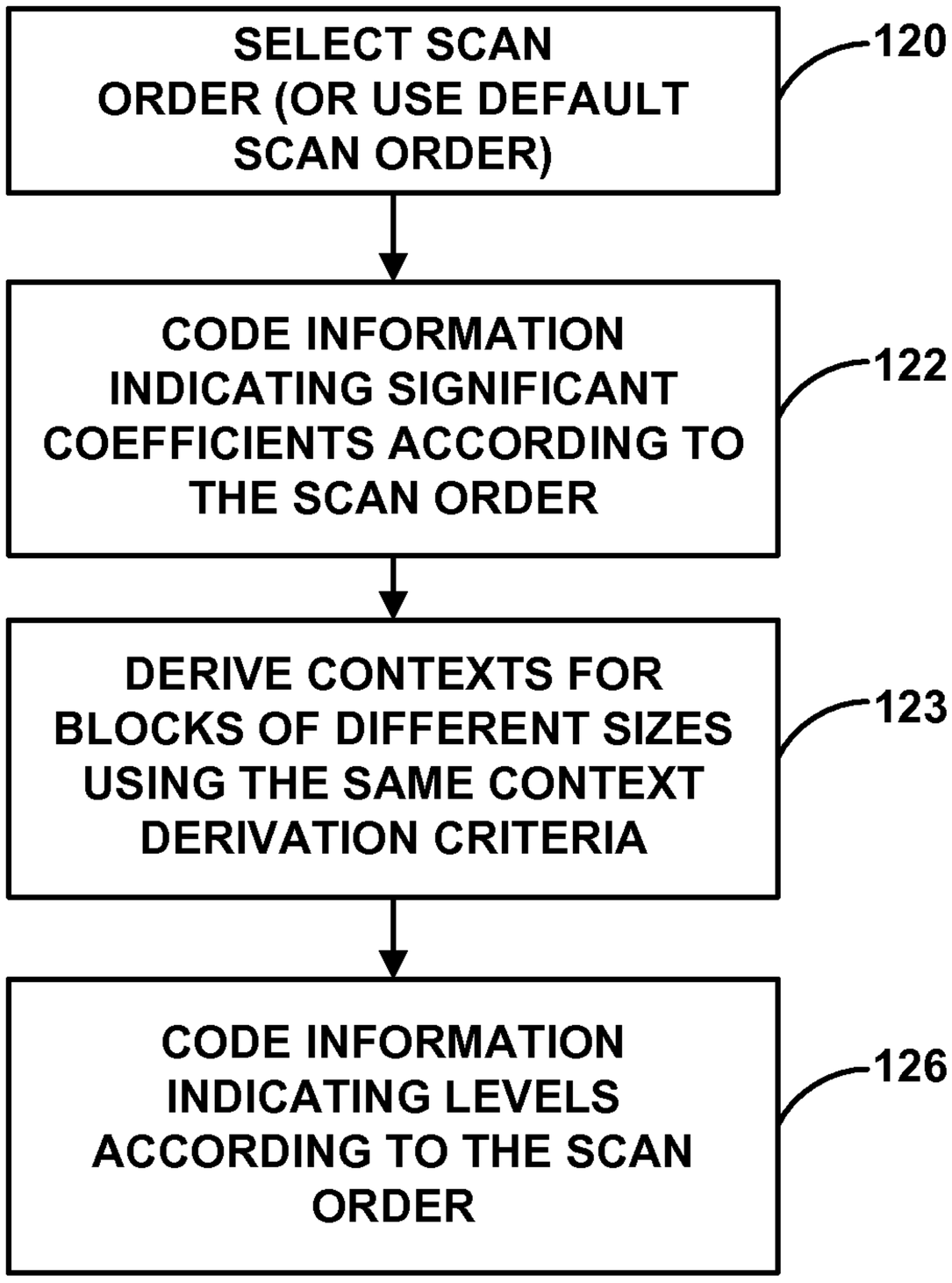

SELECT SCAN ORDER (OR USE DEFAULT SCAN ORDER)
120
CODE INFORMATION INDICATING SIGNIFICANT COEFFICIENTS ACCORDING TO THE SCAN ORDER
122
DERIVE CONTEXTS FOR BLOCKS OF DIFFERENT SIZES USING THE SAME CONTEXT DERIVATION CRITERIA
123
CODE INFORMATION INDICATING LEVELS ACCORDING TO THE SCAN ORDER
126

FIG. 23

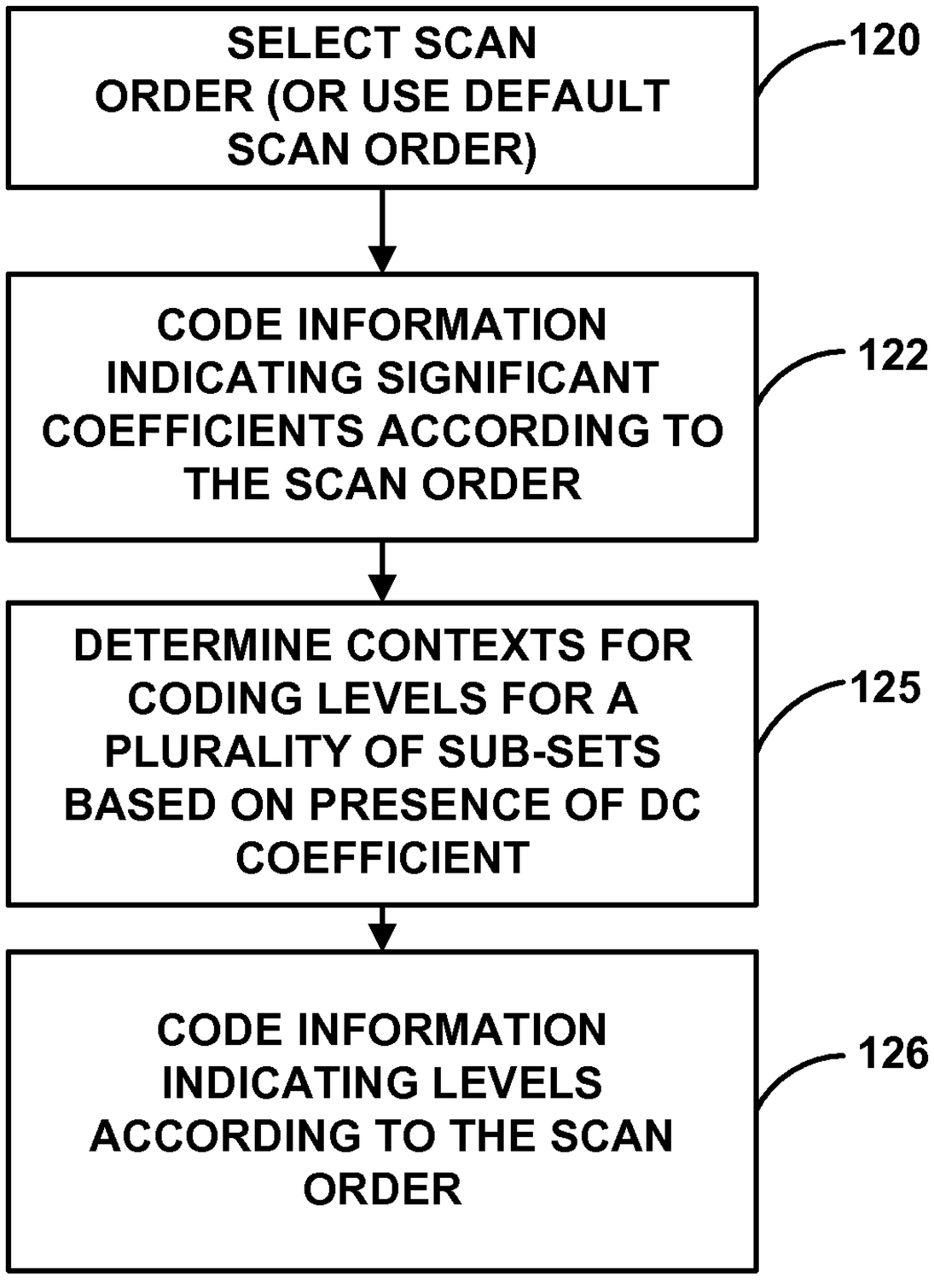

SELECT SCAN ORDER (OR USE DEFAULT SCAN ORDER)
120
CODE INFORMATION INDICATING SIGNIFICANT COEFFICIENTS ACCORDING TO THE SCAN ORDER
122
DETERMINE CONTEXTS FOR CODING LEVELS FOR A PLURALITY OF SUB-SETS BASED ON PRESENCE OF DC COEFFICIENT
125
CODE INFORMATION INDICATING LEVELS ACCORDING TO THE SCAN ORDER
126

FIG. 24

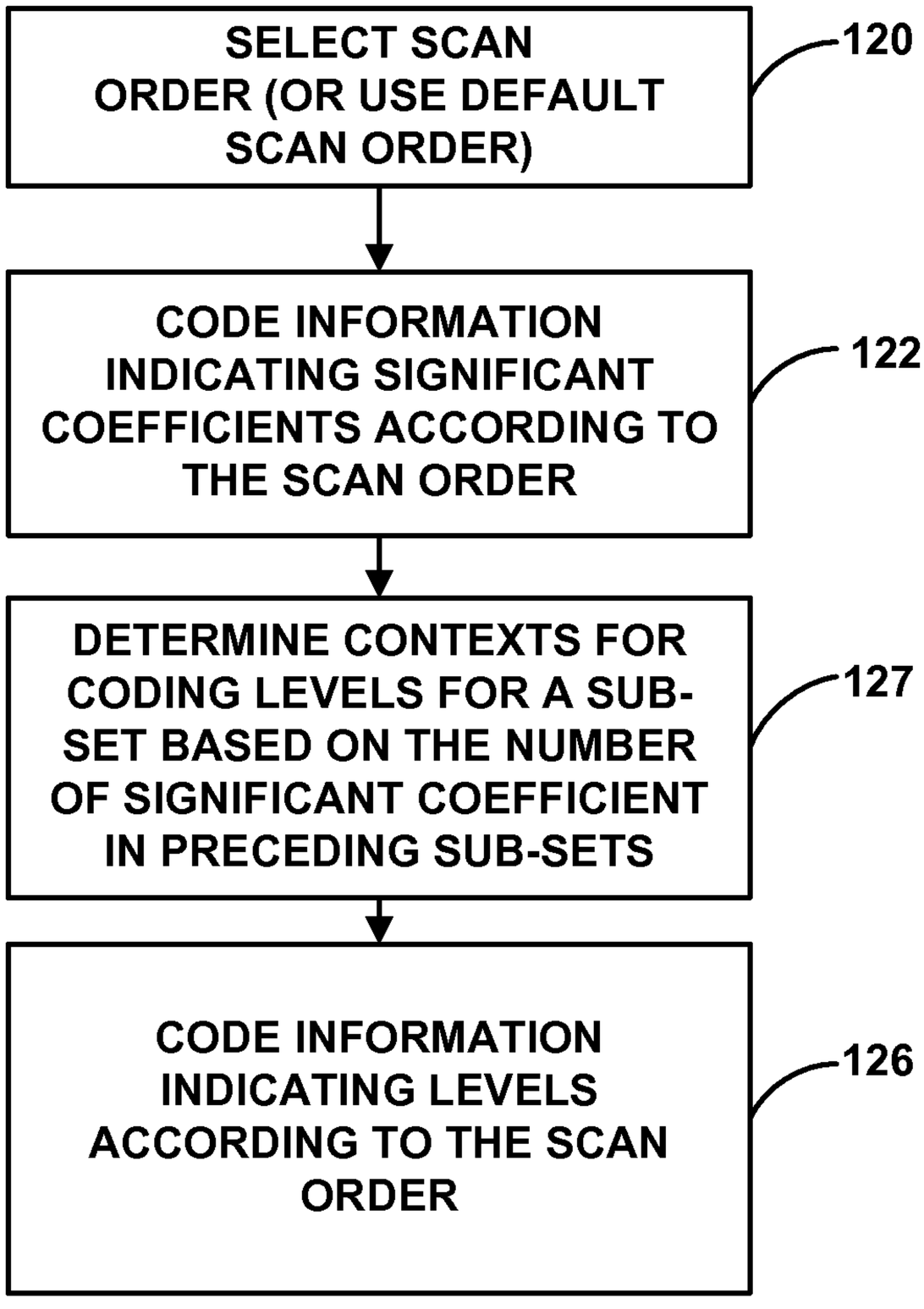

SELECT SCAN ORDER (OR USE DEFAULT SCAN ORDER)
120
CODE INFORMATION INDICATING SIGNIFICANT COEFFICIENTS ACCORDING TO THE SCAN ORDER
122
DETERMINE CONTEXTS FOR CODING LEVELS FOR A SUB-SET BASED ON THE NUMBER OF SIGNIFICANT COEFFICIENT IN PRECEDING SUB-SETS
127
CODE INFORMATION INDICATING LEVELS ACCORDING TO THE SCAN ORDER
126

FIG. 25

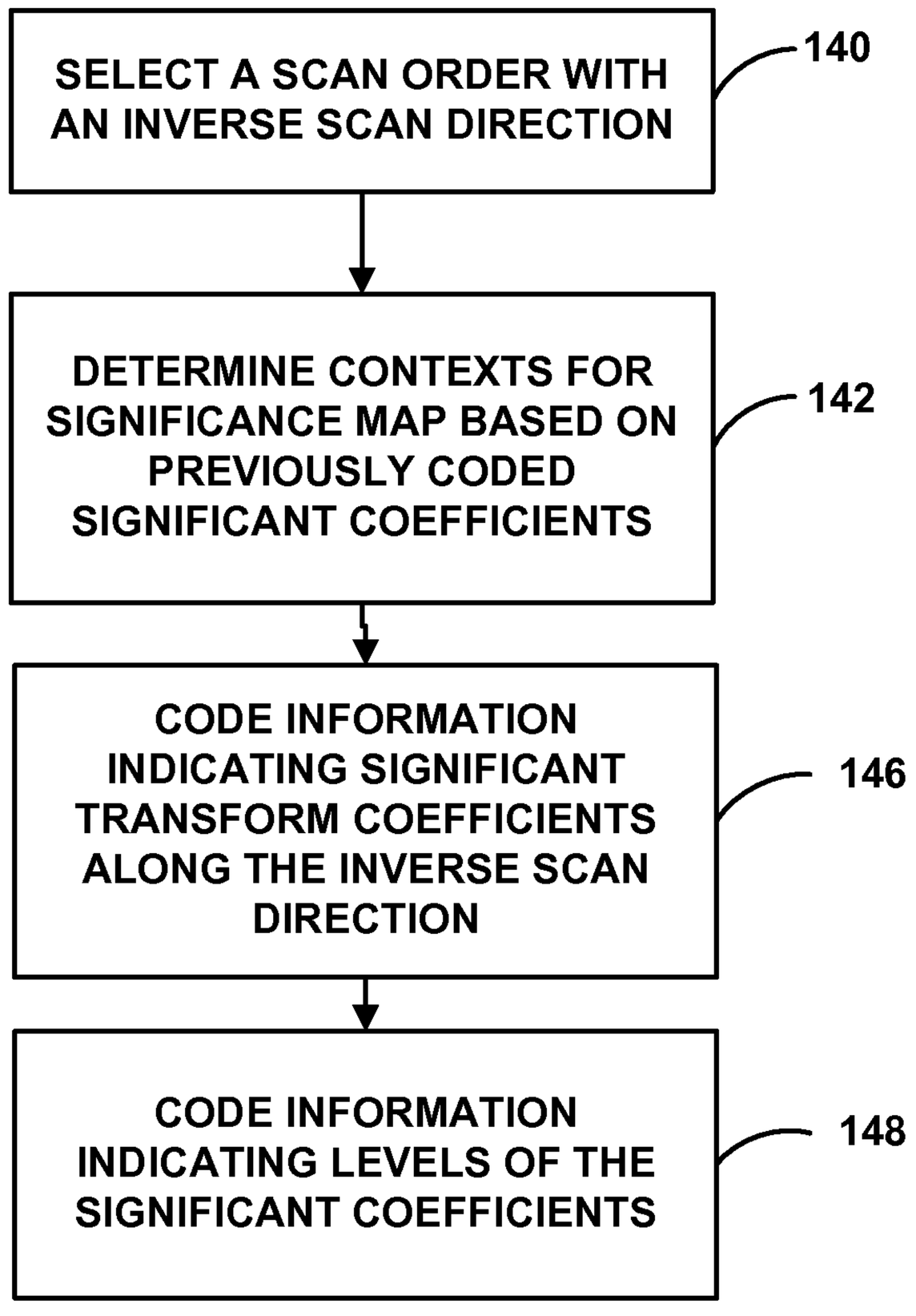

SELECT A SCAN ORDER WITH AN INVERSE SCAN DIRECTION
140
DETERMINE CONTEXTS FOR SIGNIFICANCE MAP BASED ON PREVIOUSLY CODED SIGNIFICANT COEFFICIENTS
142
CODE INFORMATION INDICATING SIGNIFICANT TRANSFORM COEFFICIENTS ALONG THE INVERSE SCAN DIRECTION
146
CODE INFORMATION INDICATING LEVELS OF THE SIGNIFICANT COEFFICIENTS
148

FIG. 26

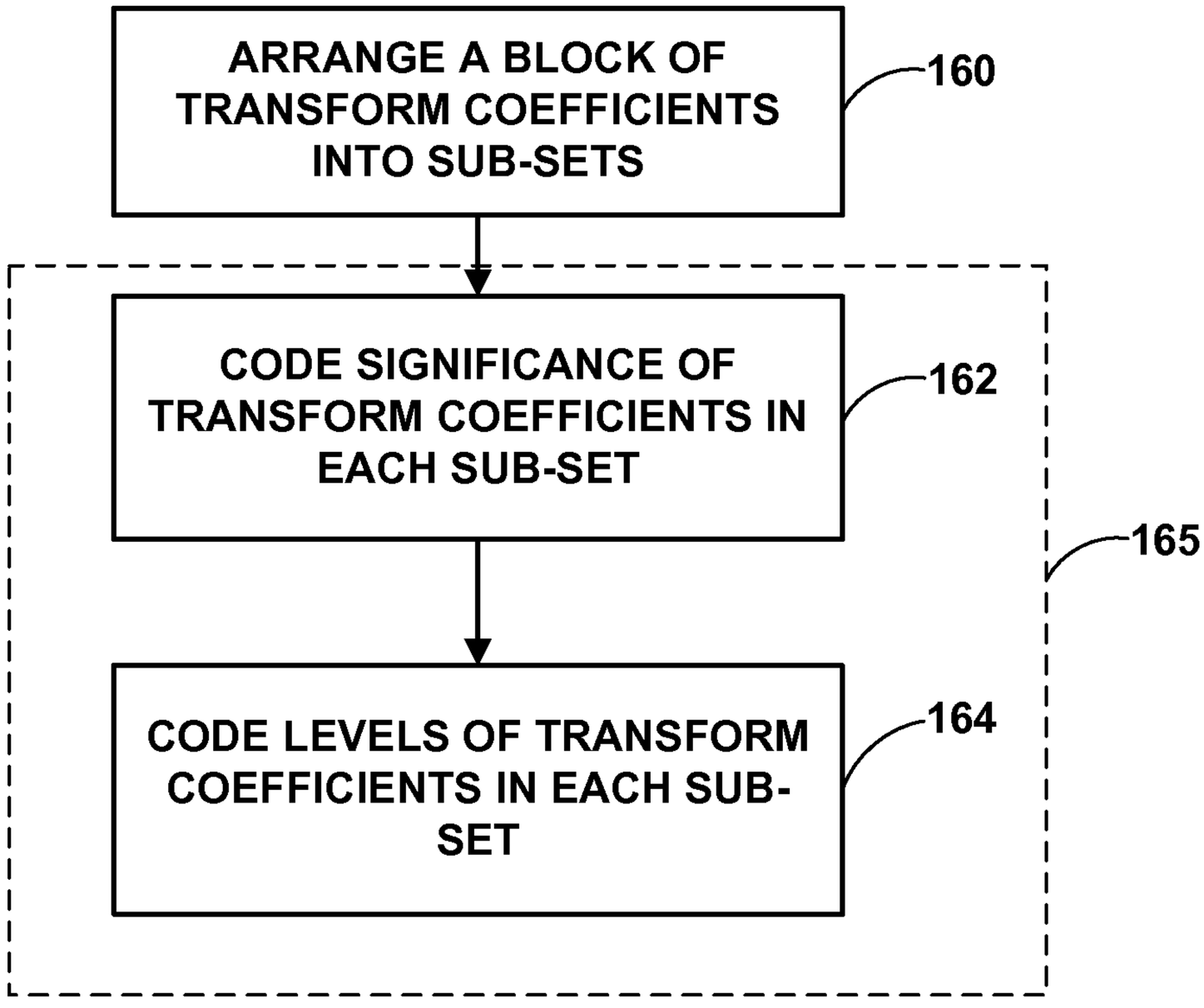

ARRANGE A BLOCK OF TRANSFORM COEFFICIENTS INTO SUB-SETS
160
CODE SIGNIFICANCE OF TRANSFORM COEFFICIENTS IN EACH SUB-SET
162
165
CODE LEVELS OF TRANSFORM COEFFICIENTS IN EACH SUB-SET
164

FIG. 27

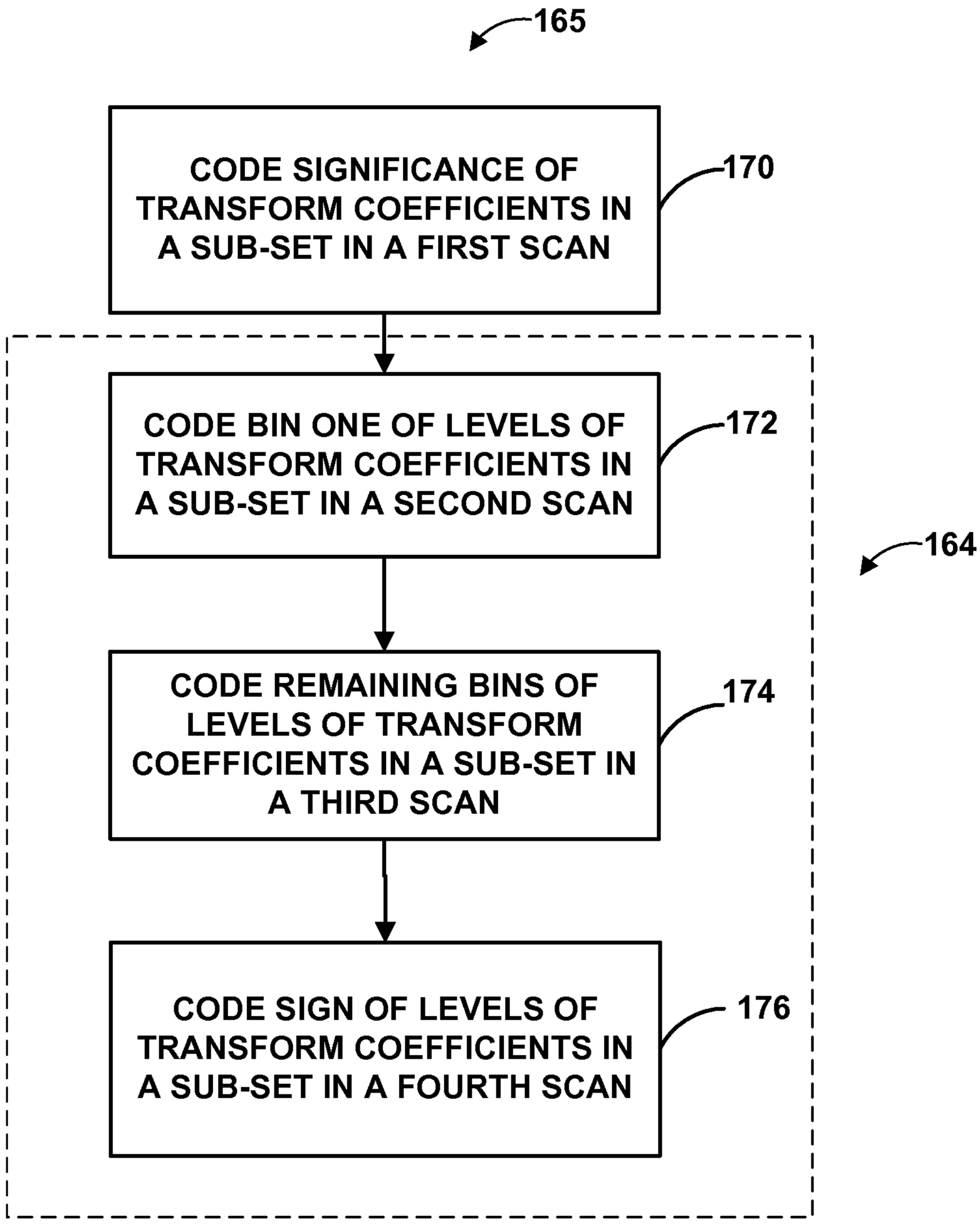

165
CODE SIGNIFICANCE OF TRANSFORM COEFFICIENTS IN A SUB-SET IN A FIRST SCAN
170
CODE BIN ONE OF LEVELS OF TRANSFORM COEFFICIENTS IN A SUB-SET IN A SECOND SCAN
172
164
CODE REMAINING BINS OF LEVELS OF TRANSFORM COEFFICIENTS IN A SUB-SET IN A THIRD SCAN
174
CODE SIGN OF LEVELS OF TRANSFORM COEFFICIENTS IN A SUB-SET IN A FOURTH SCAN
176

FIG. 28

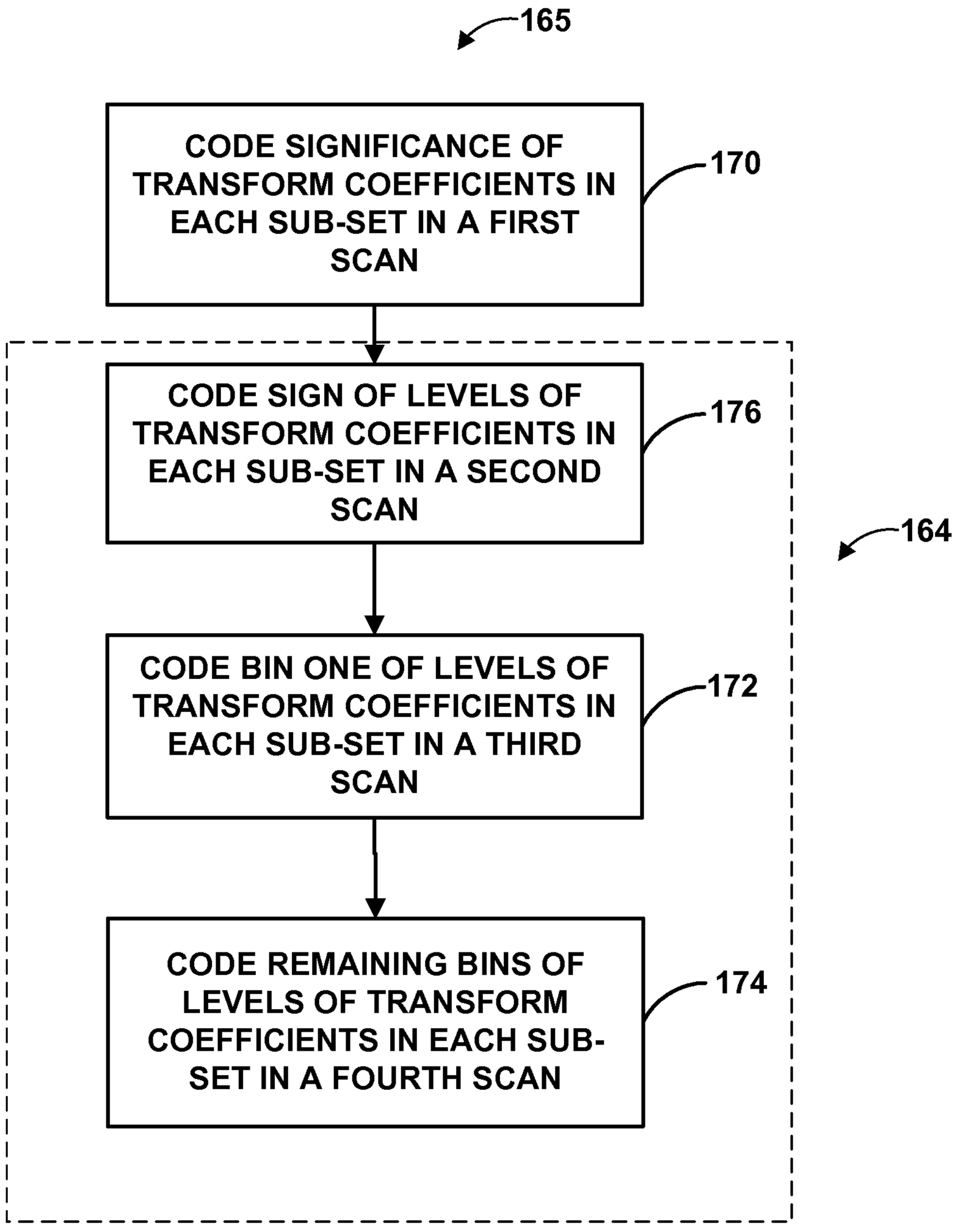

165
CODE SIGNIFICANCE OF TRANSFORM COEFFICIENTS IN EACH SUB-SET IN A FIRST SCAN
170
CODE SIGN OF LEVELS OF TRANSFORM COEFFICIENTS IN EACH SUB-SET IN A SECOND SCAN
176
164
CODE BIN ONE OF LEVELS OF TRANSFORM COEFFICIENTS IN EACH SUB-SET IN A THIRD SCAN
172
CODE REMAINING BINS OF LEVELS OF TRANSFORM COEFFICIENTS IN EACH SUB-SET IN A FOURTH SCAN
174

FIG. 29

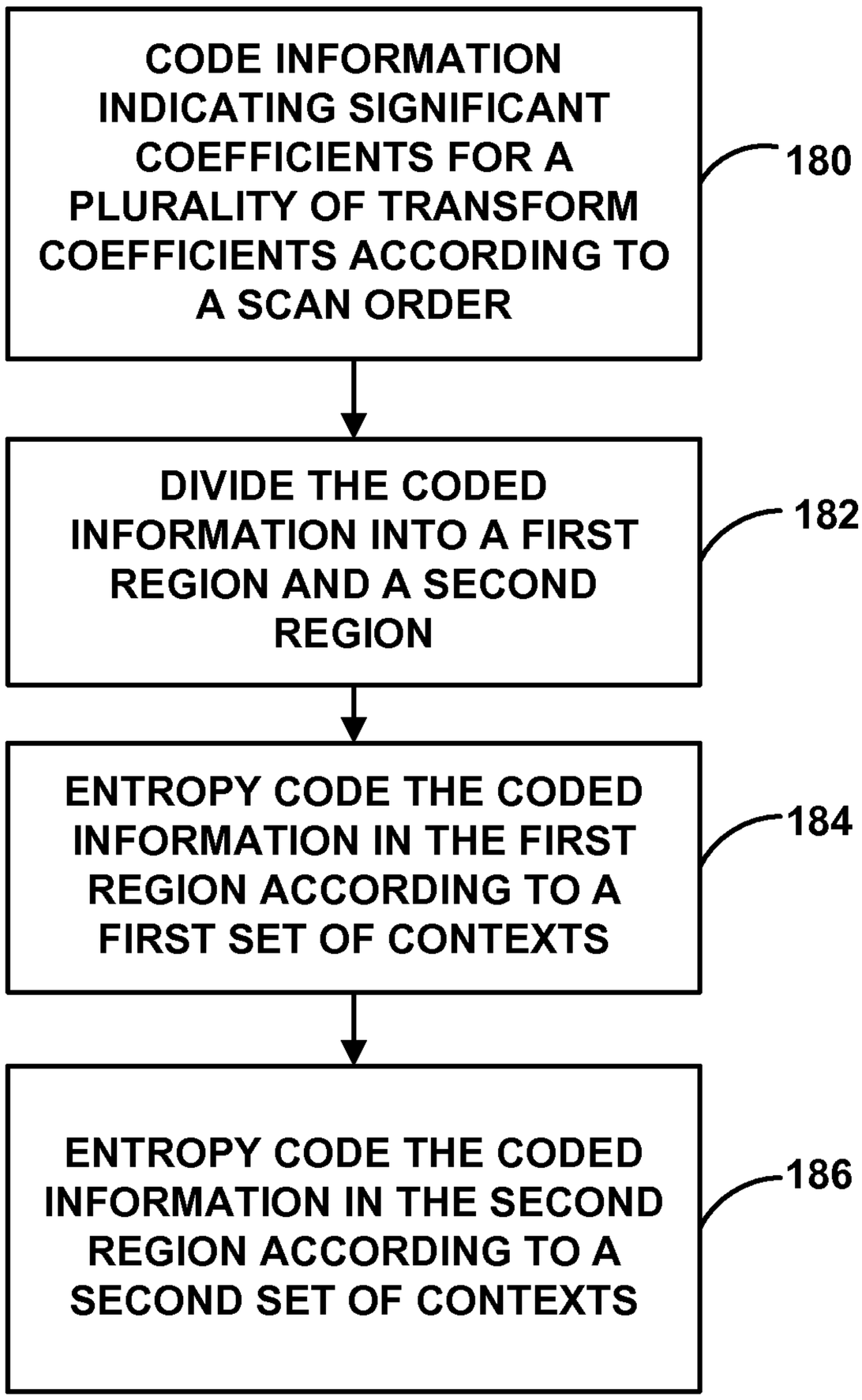
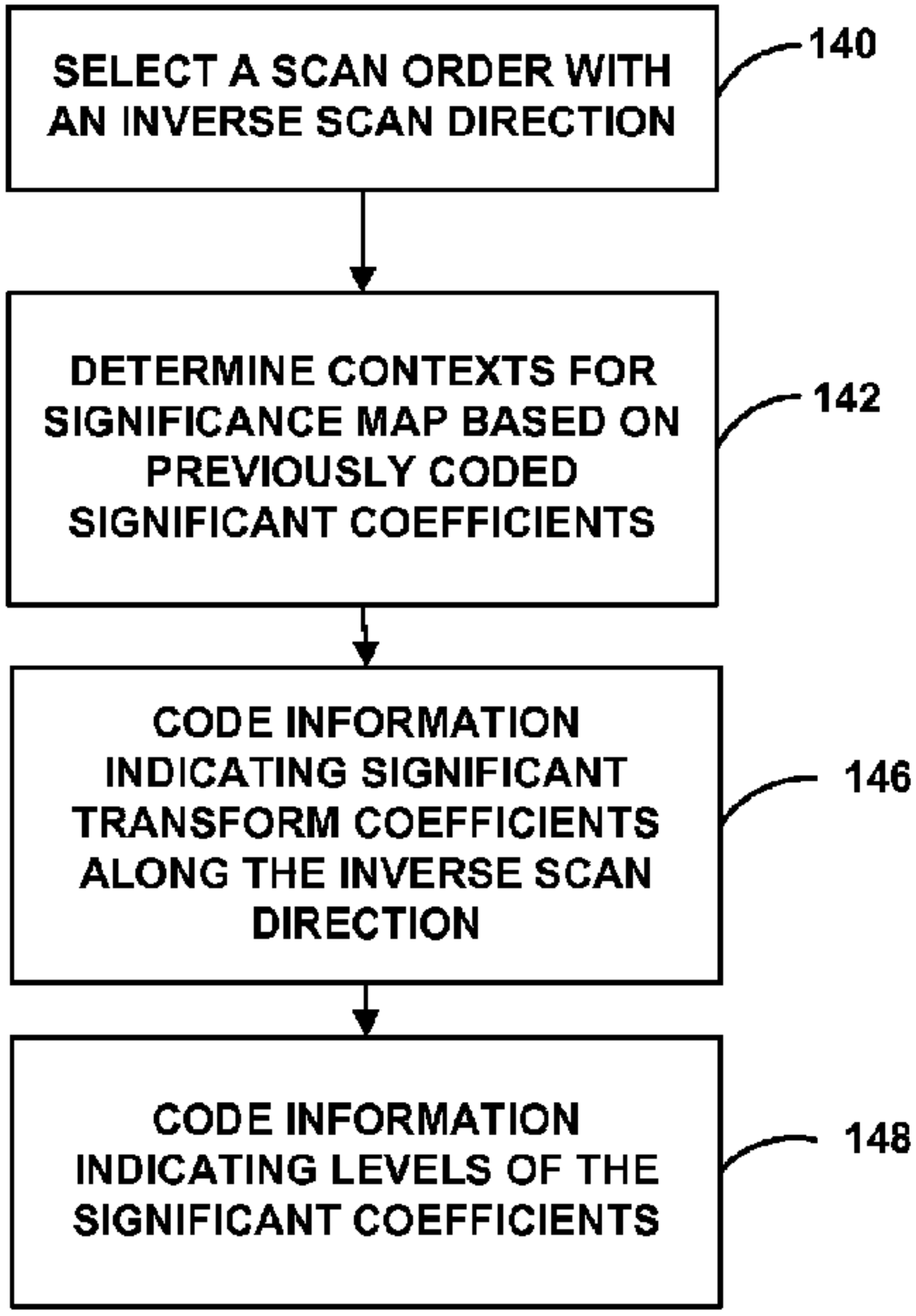

CODE INFORMATION INDICATING SIGNIFICANT COEFFICIENTS FOR A PLURALITY OF TRANSFORM COEFFICIENTS ACCORDING TO A SCAN ORDER
180
DIVIDE THE CODED INFORMATION INTO A FIRST REGION AND A SECOND REGION
182
ENTROPY CODE THE CODED INFORMATION IN THE FIRST REGION ACCORDING TO A FIRST SET OF CONTEXTS
184
ENTROPY CODE THE CODED INFORMATION IN THE SECOND REGION ACCORDING TO A SECOND SET OF CONTEXTS
186

FIG. 30


SELECT A SCAN ORDER WITH AN INVERSE SCAN DIRECTION
140
DETERMINE CONTEXTS FOR SIGNIFICANCE MAP BASED ON PREVIOUSLY CODED SIGNIFICANT COEFFICIENTS
142
CODE INFORMATION INDICATING SIGNIFICANT TRANSFORM COEFFICIENTS ALONG THE INVERSE SCAN DIRECTION
146
CODE INFORMATION INDICATING LEVELS OF THE SIGNIFICANT COEFFICIENTS
148