



(51) International Patent Classification:

H04N 19/573 (2014.01) H04N 19/176 (2014.01)

H04N 19/577 (2014.01) H04N 19/105 (2014.01)

H04N 19/119 (2014.01)

(21) International Application Number:

PCT/IB2019/054466

(22) International Filing Date:

30 May 2019 (30.05.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2018/089242

31 May 2018 (31.05.2018)

CN

PCT/CN2019/070058

02 January 2019 (02.01.2019)

CN

(71) Applicants: BEIJING BYTEDANCE NETWORK

TECHNOLOGY CO., LTD. [CN/CN]; Room B-0035, 2/

F, No.3 Building, No.30, Shixing Road, Shijingshan Dis-

trict, Beijing 100041 (CN). BYTEDANCE INC. [US/US];

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) **Inventors:** **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Hongbin**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **WANG, Yue**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN).

(74) **Agent:** **LIU, SHEN & ASSOCIATES**; 10th Floor, Building 1, 10 Caihefang Road, Haidian, District, Beijing, Beijing 100080 (CN).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

(54) Title: APPLICATION OF INTERWEAVED PREDICTION

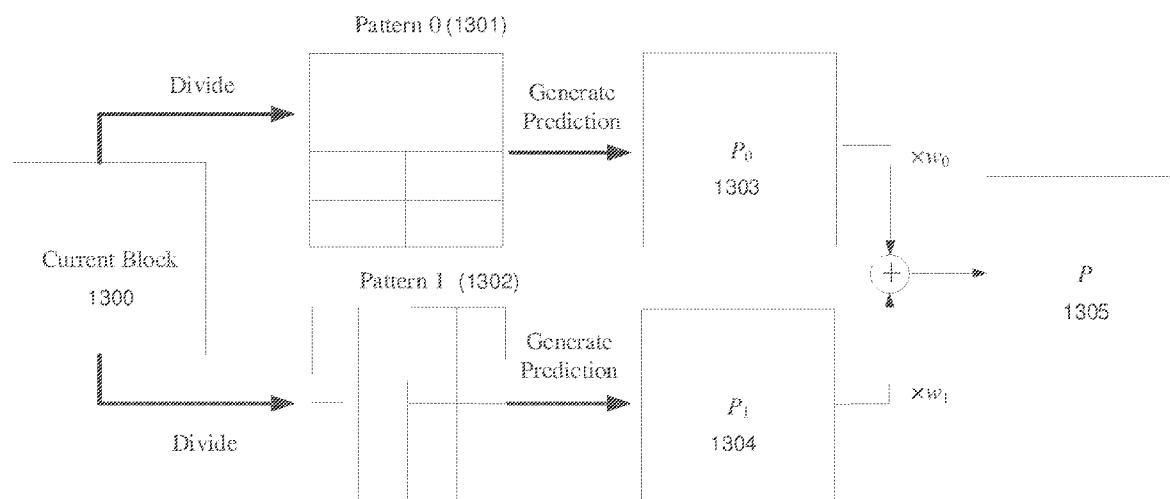


FIG. 13

(57) **Abstract:** A method of processing video blocks includes determining, due to a block meeting a condition, to apply interweaved prediction to the block, determining a prediction block based on a first intermediate prediction block and a second intermediate prediction block, and generating an encoded or a decoded representation of the block using the prediction block. The first intermediate prediction block is generated from the first set of sub-blocks obtained by partitioning the block according to a first dividing pattern and the second intermediate prediction block is generated from the second set of sub-blocks obtained by partitioning the block according to a second dividing pattern. At least one sub-block in the second set has a different size than a sub-block in the first set.



OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

- *with international search report (Art. 21(3))*
- *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

APPLICATION OF INTERWEAVED PREDICTION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priorities to and benefits of (1) prior Chinese patent application filed as International Patent Application No. PCT/CN2018/089242 on May 31, 2018, and (2) prior Chinese patent application filed as International Patent Application No. PCT/CN2019/070058 on January 2, 2019, which were subsequently abandoned after filing. For all purposes under the U.S. law, the entire disclosure of the International Patent Application Nos. PCT/CN2018/089242 and PCT/CN2019/070058 is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[0002] This patent document relates to video coding techniques, devices and systems.

BACKGROUND

[0003] Motion compensation (MC) is a technique in video processing to predict a frame in a video, given the previous and/or future frames by accounting for motion of the camera and/or objects in the video. Motion compensation can be used in the encoding of video data for video compression.

SUMMARY

[0004] This document discloses methods, systems, and devices related to sub-block based motion prediction in video motion compensation.

[0005] In one representative aspect, a method of processing video blocks is disclosed. The method includes determining, due to a block meeting a condition, to apply interweaved prediction to the block, determining a prediction block based on a first intermediate prediction block and a second intermediate prediction block, and generating an encoded or a decoded representation of the block using the prediction block. The first intermediate prediction block is generated from the first set of sub-blocks obtained by partitioning the block according to a first dividing pattern and the second intermediate prediction block is generated from the second set of

sub-blocks obtained by partitioning the block according to a second dividing pattern. At least one sub-block in the second set has a different size than a sub-block in the first set.

[0006] In another representative aspect, an apparatus comprising a processor configured to implement a method described in the present document.

5 [0007] In yet another representative aspect, the various techniques described herein may be embodied as a computer program product stored on a non-transitory computer readable media. The computer program product includes program code for carrying out the methods described herein.

[0008] In yet another representative aspect, a video decoder apparatus may implement a
10 method as described herein.

[0009] The details of one or more implementations are set forth in the drawings, the description and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

15 [0010] FIG. 1 is a schematic diagram showing an example of sub-block based prediction.

[0011] FIG. 2 shows an example of an affine motion field of a block described by two control point motion vectors.

[0012] FIG. 3 shows an example of affine motion vector field per sub-block for a block.

[0013] FIG. 4 shows an example of motion vector prediction for a block 400 in the
20 AF_INTER mode.

[0014] FIG. 5A shows an example of the selection order of candidate blocks for a current Coding Unit (CU).

[0015] FIG. 5B shows another example of candidate blocks for a current CU in the AF_MERGE mode.

25 [0016] FIG. 6 shows an example of Alternative Temporal Motion Vector Prediction (ATMVP) motion prediction process for a CU.

[0017] FIG. 7 shows an example of one CU with four sub-blocks and neighboring blocks.

[0018] FIG. 8 is a flowchart of an example method of video processing.

[0019] FIG. 9 shows an example of a functional block diagram of a video encoder or a video
30 decoder.

[0020] FIG. 10 shows an example of bilateral matching used in the Frame-Rate Up

Conversion (FRUC) method.

[0021] FIG. 11 shows an example of template matching used in the FRUC method.

[0022] FIG. 12 shows an example of unilateral Motion Estimation (ME) in the FRUC method.

5 [0023] FIG. 13 shows an example of interweaved prediction with two dividing patterns in accordance with the disclosed technology.

[0024] FIG. 14A shows an example dividing pattern in which block is divided into 4×4 sub-blocks in accordance with the disclosed technology.

10 [0025] FIG. 14B shows an example dividing pattern in which a block is divided into 8×8 sub-blocks in accordance with the disclosed technology.

[0026] FIG. 14C shows an example dividing pattern in which a block is divided into 4×8 sub-blocks in accordance with the disclosed technology.

[0027] FIG. 14D shows an example dividing pattern in which a block is divided into 8×4 sub-blocks in accordance with the disclosed technology.

15 [0028] FIG. 14E shows an example dividing pattern in which a block is divided into non-uniform sub-blocks in accordance with the disclosed technology.

[0029] FIG. 14F shows another example dividing pattern in which a block is divided into non-uniform sub-blocks in accordance with the disclosed technology.

20 [0030] FIG. 14G shows yet another example dividing pattern in which a block is divided into non-uniform sub-blocks in accordance with the disclosed technology.

[0031] FIG. 15A is an example flowchart of a method for improving bandwidth usage and prediction accuracy of a block-based motion prediction video system in accordance with the disclosed technology.

25 [0032] FIG. 15B is another example flowchart of a method for improving bandwidth usage and prediction accuracy of a block-based motion prediction video system in accordance with the disclosed technology.

[0033] FIG. 16 is a block diagram illustrating an example of the architecture for a computer system or other control device that can be utilized to implement various portions of the presently disclosed technology.

30 [0034] FIG. 17 shows a block diagram of an example embodiment of a mobile device that can be utilized to implement various portions of the presently disclosed technology.

DETAILED DESCRIPTION

[0035] Global motion compensation is one of variations of motion compensation techniques in video compression and can be used for predicting camera's motion. However, moving objects within a frame of a video file are not sufficiently represented by various implementations of the global motion compensation. Local motion estimation, such as block motion compensation, in which the frames are partitioned in blocks of pixels for performing the motion prediction, can be used to account for the objects moving within the frames.

[0036] Sub-block based prediction, which was developed based on the block motion compensation, was first introduced into the video coding standard by High Efficiency Video Coding (HEVC) Annex I (3D-HEVC).

[0037] FIG. 1 is a schematic diagram showing an example of sub-block based prediction. With sub-block based prediction, a block 100, such as a Coding Unit (CU) or a Prediction Unit (PU), is divided into several non-overlapped sub-blocks 101. Different sub-blocks may be assigned different motion information, such as reference index or Motion Vector (MV). Motion compensation is then performed individually for each sub-block.

[0038] To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded jointly by the Video Coding Expert Group (VCEG) and the Moving Picture Expert Group (MPEG) in 2015. Many methods have been adopted by JVET and added into the reference software named Joint Exploration Model (JEM). In JEM, sub-block based prediction is adopted in several coding techniques, such as affine prediction, Alternative temporal motion vector prediction (ATMVP), spatial-temporal motion vector prediction (STMVP), Bi-directional Optical flow (BIO), and Frame-Rate Up Conversion (FRUC), which are discussed in detail below.

[0039] Affine Prediction

[0040] In HEVC, only translation motion model is applied for motion compensation prediction (MCP). However, the camera and objects may have many kinds of motion, e.g. zoom in/out, rotation, perspective motions, and/or other irregular motions. JEM, on the other hand, applies a simplified affine transform motion compensation prediction.

[0041] FIG. 2 shows an example of an affine motion field of a block 200 described by two control point motion vectors V_0 and V_1 . The motion vector field (MVF) of the block 200 can be

described by the following equation:

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

[0042] As shown in FIG. 2, (v_{0x}, v_{0y}) is the motion vector of the top-left corner control point, and (v_{1x}, v_{1y}) is the motion vector of the top-right corner control point. To simplify the motion compensation prediction, sub-block based affine transform prediction can be applied. The sub-block size $M \times N$ is derived as follows:

$$\begin{cases} M = \text{clip3}(4, w, \frac{w \times MvPre}{\max(\text{abs}(v_{1x} - v_{0x}), \text{abs}(v_{1y} - v_{0y}))}) \\ N = \text{clip3}(4, h, \frac{h \times MvPre}{\max(\text{abs}(v_{2x} - v_{0x}), \text{abs}(v_{2y} - v_{0y}))}) \end{cases} \quad \text{Eq. (2)}$$

[0043] Here, $MvPre$ is the motion vector fraction accuracy (e.g., 1/16 in JEM). (v_{2x}, v_{2y}) is the motion vector of the bottom-left control point, calculated according to Eq. (1). M and N can be adjusted downward if necessary to make it a divisor of w and h , respectively.

[0044] FIG. 3 shows an example of affine MVF per sub-block for a block 300. To derive motion vector of each $M \times N$ sub-block, the motion vector of the center sample of each sub-block can be calculated according to Eq. (1), and rounded to the motion vector fraction accuracy (e.g., 1/16 in JEM). Then the motion compensation interpolation filters can be applied to generate the prediction of each sub-block with derived motion vector. After the MCP, the high accuracy motion vector of each sub-block is rounded and saved as the same accuracy as the normal motion vector.

[0045] In the JEM, there are two affine motion modes: AF_INTER mode and AF_MERGE mode. For CUs with both width and height larger than 8, AF_INTER mode can be applied. An affine flag in CU level is signaled in the bitstream to indicate whether AF_INTER mode is used. In the AF_INTER mode, a candidate list with motion vector pair $\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$ is constructed using the neighboring blocks.

[0046] FIG. 4 shows an example of motion vector prediction (MVP) for a block 400 in the AF_INTER mode. As shown in FIG. 4, v_0 is selected from the motion vectors of the sub-block A, B, or C. The motion vectors from the neighboring blocks can be scaled according to the reference list. The motion vectors can also be scaled according to the relationship among the Picture Order Count (POC) of the reference for the neighboring block, the POC of the reference

for the current CU, and the POC of the current CU. The approach to select v_1 from the neighboring sub-block D and E is similar. When the number of candidate list is smaller than 2, the list is padded by the motion vector pair composed by duplicating each of the AMVP candidates. When the candidate list is larger than 2, the candidates can be firstly sorted according to the neighboring motion vectors (e.g., based on the similarity of the two motion vectors in a pair candidate). In some implementations, the first two candidates are kept. In some embodiments, a Rate Distortion (RD) cost check is used to determine which motion vector pair candidate is selected as the control point motion vector prediction (CPMVP) of the current CU. An index indicating the position of the CPMVP in the candidate list can be signaled in the bitstream. After the CPMVP of the current affine CU is determined, affine motion estimation is applied and the control point motion vector (CPMV) is found. Then the difference of the CPMV and the CPMVP is signaled in the bitstream.

[0047] When a CU is applied in AF_MERGE mode, it gets the first block coded with an affine mode from the valid neighboring reconstructed blocks. FIG. 5A shows an example of the selection order of candidate blocks for a current CU 500. As shown in FIG. 5A, the selection order can be from left (501), above (502), above right (503), left bottom (504) to above left (505) of the current CU 500. FIG. 5B shows another example of candidate blocks for a current CU 500 in the AF_MERGE mode. If the neighboring left bottom block 501 is coded in affine mode, as shown in FIG. 5B, the motion vectors v_2 , v_3 and v_4 of the top left corner, above right corner, and left bottom corner of the CU containing the sub-block 501 are derived. The motion vector v_0 of the top left corner on the current CU 500 is calculated based on v_2 , v_3 and v_4 . The motion vector v_1 of the above right of the current CU can be calculated accordingly.

[0048] After the CPMV of the current CU v_0 and v_1 are computed according to the affine motion model in Eq. (1), the MVF of the current CU can be generated. In order to identify whether the current CU is coded with AF_MERGE mode, an affine flag can be signaled in the bitstream when there is at least one neighboring block is coded in affine mode.

[0049] Alternative Temporal Motion Vector Prediction (ATMVP)

[0050] In the ATMVP method, the temporal motion vector prediction (TMVP) method is modified by fetching multiple sets of motion information (including motion vectors and reference indices) from blocks smaller than the current CU.

[0051] FIG. 6 shows an example of ATMVP motion prediction process for a CU 600. The

ATMVP method predicts the motion vectors of the sub-CUs 601 within a CU 600 in two steps. The first step is to identify the corresponding block 651 in a reference picture 650 with a temporal vector. The reference picture 650 is also referred to as the motion source picture. The second step is to split the current CU 600 into sub-CUs 601 and obtain the motion vectors as well as the reference indices of each sub-CU from the block corresponding to each sub-CU.

[0052] In the first step, a reference picture 650 and the corresponding block is determined by the motion information of the spatial neighboring blocks of the current CU 600. To avoid the repetitive scanning process of neighboring blocks, the first merge candidate in the merge candidate list of the current CU 600 is used. The first available motion vector as well as its associated reference index are set to be the temporal vector and the index to the motion source picture. This way, the corresponding block may be more accurately identified, compared with TMVP, wherein the corresponding block (sometimes called collocated block) is always in a bottom-right or center position relative to the current CU.

[0053] In the second step, a corresponding block of the sub-CU 651 is identified by the temporal vector in the motion source picture 650, by adding to the coordinate of the current CU the temporal vector. For each sub-CU, the motion information of its corresponding block (e.g., the smallest motion grid that covers the center sample) is used to derive the motion information for the sub-CU. After the motion information of a corresponding $N \times N$ block is identified, it is converted to the motion vectors and reference indices of the current sub-CU, in the same way as TMVP of HEVC, wherein motion scaling and other procedures apply. For example, the decoder checks whether the low-delay condition (e.g. the POCs of all reference pictures of the current picture are smaller than the POC of the current picture) is fulfilled and possibly uses motion vector MV_x (e.g., the motion vector corresponding to reference picture list X) to predict motion vector MV_y (e.g., with X being equal to 0 or 1 and Y being equal to $1-X$) for each sub-CU.

[0054] Spatial Temporal Motion Vector Prediction (STMVP)

[0055] In the STMVP method, the motion vectors of the sub-CUs are derived recursively, following raster scan order. FIG. 7 shows an example of one CU with four sub-blocks and neighboring blocks. Consider an 8×8 CU 700 that includes four 4×4 sub-CUs A (701), B (702), C (703), and D (704). The neighboring 4×4 blocks in the current frame are labelled as a (711), b (712), c (713), and d (714).

[0056] The motion derivation for sub-CU A starts by identifying its two spatial neighbors.

The first neighbor is the $N \times N$ block above sub-CU A 701 (block c 713). If this block c (713) is not available or is intra coded the other $N \times N$ blocks above sub-CU A (701) are checked (from left to right, starting at block c 713). The second neighbor is a block to the left of the sub-CU A 701 (block b 712). If block b (712) is not available or is intra coded other blocks to the left of sub-CU A 701 are checked (from top to bottom, starting at block b 712). The motion information obtained from the neighboring blocks for each list is scaled to the first reference frame for a given list. Next, temporal motion vector predictor (TMVP) of sub-block A 701 is derived by following the same procedure of TMVP derivation as specified in HEVC. The motion information of the collocated block at block D 704 is fetched and scaled accordingly. Finally, after retrieving and scaling the motion information, all available motion vectors are averaged separately for each reference list. The averaged motion vector is assigned as the motion vector of the current sub-CU.

[0057] Frame Rate Up-Conversion (FRUC)

[0058] A FRUC flag can be signaled for a CU when its merge flag is true. When the FRUC flag is false, a merge index can be signaled and the regular merge mode is used. When the FRUC flag is true, an additional FRUC mode flag can be signaled to indicate which method (e.g., bilateral matching or template matching) is to be used to derive motion information for the block.

[0059] At encoder side, the decision on whether using FRUC merge mode for a CU is based on RD cost selection as done for normal merge candidate. For example, multiple matching modes (e.g., bilateral matching and template matching) are checked for a CU by using RD cost selection. The one leading to the minimal cost is further compared to other CU modes. If a FRUC matching mode is the most efficient one, FRUC flag is set to true for the CU and the related matching mode is used.

[0060] Typically, motion derivation process in FRUC merge mode has two steps: a CU-level motion search is first performed, then followed by a Sub-CU level motion refinement. At CU level, an initial motion vector is derived for the whole CU based on bilateral matching or template matching. First, a list of MV candidates is generated and the candidate that leads to the minimum matching cost is selected as the starting point for further CU level refinement. Then a local search based on bilateral matching or template matching around the starting point is performed. The MV results in the minimum matching cost is taken as the MV for the whole CU.

Subsequently, the motion information is further refined at sub-CU level with the derived CU motion vectors as the starting points.

[0061] For example, the following derivation process is performed for a $W \times H$ CU motion information derivation. At the first stage, MV for the whole $W \times H$ CU is derived. At the second stage, the CU is further split into $M \times M$ sub-CUs. The value of M is calculated as in (16), D is a predefined splitting depth which is set to 3 by default in the JEM. Then the MV for each sub-CU is derived.

$$M = \max\{4, \min\{\frac{M}{2^D}, \frac{N}{2^D}\}\} \quad \text{Eq. (13)}$$

[0062] FIG. 10 shows an example of bilateral matching used in the Frame-Rate Up

Conversion (FRUC) method. The bilateral matching is used to derive motion information of the current CU by finding the closest match between two blocks along the motion trajectory of the current CU (1000) in two different reference pictures (1010, 1011). Under the assumption of continuous motion trajectory, the motion vectors MV0 (1001) and MV1 (1002) pointing to the two reference blocks are proportional to the temporal distances, e.g., TD0 (1003) and TD1 (1004), between the current picture and the two reference pictures. In some embodiments, when the current picture 1000 is temporally between the two reference pictures (1010, 1011) and the temporal distance from the current picture to the two reference pictures is the same, the bilateral matching becomes mirror based bi-directional MV.

[0063] FIG. 11 shows an example of template matching used in the FRUC method.

Template matching can be used to derive motion information of the current CU 1100 by finding the closest match between a template (e.g., top and/or left neighboring blocks of the current CU) in the current picture and a block (e.g., same size to the template) in a reference picture 1110. Except the aforementioned FRUC merge mode, the template matching can also be applied to AMVP mode. In both JEM and HEVC, AMVP has two candidates. With the template matching method, a new candidate can be derived. If the newly derived candidate by template matching is different to the first existing AMVP candidate, it is inserted at the very beginning of the AMVP candidate list and then the list size is set to two (e.g., by removing the second existing AMVP candidate). When applied to AMVP mode, only CU level search is applied.

[0064] The MV candidate set at CU level can include the following: (1) original AMVP candidates if the current CU is in AMVP mode, (2) all merge candidates, (3) several MVs in the interpolated MV field (described later), and top and left neighboring motion vectors.

[0065] When using bilateral matching, each valid MV of a merge candidate can be used as an input to generate a MV pair with the assumption of bilateral matching. For example, one valid MV of a merge candidate is (MV_a, ref_a) at reference list A. Then the reference picture ref_b of its paired bilateral MV is found in the other reference list B so that ref_a and ref_b are temporally at different sides of the current picture. If such a ref_b is not available in reference list B, ref_b is determined as a reference which is different from ref_a and its temporal distance to the current picture is the minimal one in list B. After ref_b is determined, MV_b is derived by scaling MV_a based on the temporal distance between the current picture and ref_a, ref_b .

[0066] In some implementations, four MVs from the interpolated MV field can also be added to the CU level candidate list. More specifically, the interpolated MVs at the position $(0, 0)$, $(W/2, 0)$, $(0, H/2)$ and $(W/2, H/2)$ of the current CU are added. When FRUC is applied in AMVP mode, the original AMVP candidates are also added to CU level MV candidate set. In some implementations, at the CU level, 15 MVs for AMVP CUs and 13 MVs for merge CUs can be added to the candidate list.

[0067] The MV candidate set at sub-CU level includes an MV determined from a CU-level search, (2) top, left, top-left and top-right neighboring MVs, (3) scaled versions of collocated MVs from reference pictures, (4) one or more ATMVP candidates (e.g., up to four), and (5) one or more STMVP candidates (e.g., up to four). The scaled MVs from reference pictures are derived as follows. The reference pictures in both lists are traversed. The MVs at a collocated position of the sub-CU in a reference picture are scaled to the reference of the starting CU-level MV. ATMVP and STMVP candidates can be the four first ones. At the sub-CU level, one or more MVs (e.g., up to 17) are added to the candidate list.

[0068] *Generation of interpolated MV field*

[0069] Before coding a frame, interpolated motion field is generated for the whole picture based on unilateral ME. Then the motion field may be used later as CU level or sub-CU level MV candidates.

[0070] In some embodiments, the motion field of each reference pictures in both reference lists is traversed at 4×4 block level. FIG. 12 shows an example of unilateral Motion Estimation (ME) 1200 in the FRUC method. For each 4×4 block, if the motion associated to the block passing through a 4×4 block in the current picture and the block has not been assigned any interpolated motion, the motion of the reference block is scaled to the current picture according

to the temporal distance TD0 and TD1 (the same way as that of MV scaling of TMVP in HEVC) and the scaled motion is assigned to the block in the current frame. If no scaled MV is assigned to a 4×4 block, the block's motion is marked as unavailable in the interpolated motion field.

[0071] *Interpolation and matching cost*

5 **[0072]** When a motion vector points to a fractional sample position, motion compensated interpolation is needed. To reduce complexity, bi-linear interpolation instead of regular 8-tap HEVC interpolation can be used for both bilateral matching and template matching.

[0073] The calculation of matching cost is a bit different at different steps. When selecting the candidate from the candidate set at the CU level, the matching cost can be the absolute sum difference (SAD) of bilateral matching or template matching. After the starting MV is
10 determined, the matching cost C of bilateral matching at sub-CU level search is calculated as follows:

$$C = SAD + w \cdot (|MV_x - MV_x^s| + |MV_y - MV_y^s|) \quad \text{Eq. (14)}$$

[0074] Here, w is a weighting factor. In some embodiments, w can be empirically set to 4.
15 MV and MV^s indicate the current MV and the starting MV, respectively. SAD may still be used as the matching cost of template matching at sub-CU level search.

[0075] In FRUC mode, MV is derived by using luma samples only. The derived motion will be used for both luma and chroma for MC inter prediction. After MV is decided, final MC is performed using 8-taps interpolation filter for luma and 4-taps interpolation filter for chroma.

20 **[0076]** MV refinement is a pattern based MV search with the criterion of bilateral matching cost or template matching cost. In the JEM, two search patterns are supported – an unrestricted center-biased diamond search (UCBDS) and an adaptive cross search for MV refinement at the CU level and sub-CU level, respectively. For both CU and sub-CU level MV refinement, the MV is directly searched at quarter luma sample MV accuracy, and this is followed by one-eighth
25 luma sample MV refinement. The search range of MV refinement for the CU and sub-CU step are set equal to 8 luma samples.

[0077] In the bilateral matching merge mode, bi-prediction is applied because the motion information of a CU is derived based on the closest match between two blocks along the motion trajectory of the current CU in two different reference pictures. In the template matching merge
30 mode, the encoder can choose among uni-prediction from list0, uni-prediction from list1, or bi-prediction for a CU. The selection can be based on a template matching cost as follows:

[0078] If $\text{costBi} \leq \text{factor} * \min(\text{cost0}, \text{cost1})$

[0079] bi-prediction is used;

[0080] Otherwise, if $\text{cost0} \leq \text{cost1}$

[0081] uni-prediction from list0 is used;

5 [0082] Otherwise,

[0083] uni-prediction from list1 is used;

[0084] Here, cost0 is the SAD of list0 template matching, cost1 is the SAD of list1 template matching and costBi is the SAD of bi-prediction template matching. For example, when the value of factor is equal to 1.25, it means that the selection process is biased toward bi-prediction.

10 The inter prediction direction selection can be applied to the CU-level template matching process.

[0085] The sub-block based prediction techniques discussed above can be used to obtain more accurate motion information of each sub-block when the size of sub-blocks is smaller. However, smaller sub-blocks impose a higher bandwidth requirement in motion compensation.

15 On the other hand, motion information derived for smaller sub-block may not be accurate, especially when there are some noises in a block. Therefore, having a fixed sub-block size within one block may be suboptimal.

[0086] This document describes techniques that can be used in various embodiments to use non-uniform and/or variable sub-block sizes to address the bandwidth and accuracy problems that a fixed sub-block size introduces. The techniques, also referred to as interweaved prediction, use different ways of dividing a block so that motion information can be obtained more robustly without increasing bandwidth consumption.

[0087] Using the interweaved prediction techniques, a block is divided into sub-blocks with one or more dividing patterns. A dividing pattern represents the way to divide a block into sub-blocks, including the size of sub-blocks and the position of sub-blocks. For each dividing pattern, a corresponding prediction block may be generated by deriving motion information of each sub-block based on the dividing pattern. Therefore, in some embodiments, multiple prediction blocks may be generated by multiple dividing patterns even for one prediction direction. In some embodiments, for each prediction direction, only one dividing pattern may be applied.

[0088] FIG. 13 shows an example of interweaved prediction with two dividing patterns in

accordance with the disclosed technology. A current block 1300 can be divided into multiple patterns. For example, as shown in FIG. 13, the current block is divided into both Pattern 0 (1301) and Pattern 1 (1302). Two prediction blocks, P_0 (1303) and P_1 (1304), are generated. A final prediction block P (1305) of the current block 1300 can be generated by computing a weighted sum of P_0 (1303) and P_1 (1304).

[0089] More generally, given X dividing patterns, X prediction blocks of the current block, denoted as $P_0, P_1, \dots, P_{X-1}$, can be generated by sub-block based prediction with the X dividing patterns. The final prediction of the current block, denoted as P , can be generated as

$$P(x, y) = \frac{\sum_{i=0}^{X-1} w_i(x, y) \times P_i(x, y)}{\sum_{i=0}^{X-1} w_i(x, y)} \quad \text{Eq. (15)}$$

[0090] Here, (x, y) is the coordinate of a pixel in the block and $w_i(x, y)$ is the weighting value of P_i . By the way of example, and not by limitation, the weights can be expressed as:

$$\sum_{i=0}^{X-1} w_i(x, y) = (1 \ll N) \quad \text{Eq. (16)}$$

[0091] N is a non-negative value. Alternatively, the bit-shifting operation in Eq. (16) can also be expressed as:

$$\sum_{i=0}^{X-1} w_i(x, y) = 2^N \quad \text{Eq. (17)}$$

[0092] The sum of the weights being a power of two allows a more efficient computation of the weighted sum P by performing a bit-shifting operation instead of a floating-point division.

[0093] Dividing patterns can have different shapes, or sizes, or positions of sub-blocks. In some embodiments, a dividing pattern may include irregular sub-block sizes. FIGS. 14A-G

show several examples of dividing patterns for a 16×16 block. In FIG. 14A, a block is divided into 4×4 sub-blocks in accordance with the disclosed technology. This pattern is also used in JEM. FIG. 14B shows an example of a block being divided into 8×8 sub-blocks in accordance with the disclosed technology. FIG. 14C shows an example of the block being divided into 8×4 sub-blocks in accordance with the disclosed technology. FIG. 14D shows an example of the

block being divided into 4×8 sub-blocks in accordance with the disclosed technology. In FIG. 14E, a portion of the block is divided into 4×4 sub-blocks in accordance with the disclosed technology. The pixels at block boundaries are divided in smaller sub-blocks with sizes like 2×4 , 4×2 or 2×2 . Some sub-blocks may be merged to form larger sub-blocks. FIG. 14F shows an example of adjacent sub-blocks, such as 4×4 sub-blocks and 2×4 sub-blocks, that are merged

to form larger sub-blocks with sizes like 6×4 , 4×6 or 6×6 . In FIG. 14G, a portion of the block is

divided into 8×8 sub-blocks. The pixels at block boundaries are divided in smaller sub-blocks with sizes like 8×4 , 4×8 or 4×4 instead.

[0094] The shapes and sizes of sub-blocks in sub-block based prediction can be determined based on the shape and/or size of the coding block and/or coded block information. Coded block information can include a type of coding algorithm used on the block and/or sub-block, such as whether the motion compensation prediction is (1) an affine prediction method, (2) an alternative temporal motion vector prediction method, (3) a spatial-temporal motion vector prediction method, (4) a bi-directional optical flow method, or (5) a frame-rate up conversion method. For example, in some embodiments, the sub-blocks have a size of $4 \times N$ (or $8 \times N$, etc.) when the current block has a size of $M \times N$. That is, the sub-blocks have the same height as the current block. In some embodiments, the sub-blocks have a size of $M \times 4$ (or $M \times 8$, etc.) when the current block has a size of $M \times N$. That is, the sub-blocks have the same width as the current block. In some embodiments, the sub-blocks have a size of $A \times B$ with $A > B$ (e.g., 8×4) when the current block has a size of $M \times N$, where $M > N$. Alternatively, the sub-blocks can have the size of $B \times A$ (e.g. 4×8).

[0095] In some embodiments, the current block has a size of $M \times N$. The sub-blocks have a size of $A \times B$ when $M \times N \leq T$ (or $\text{Min}(M, N) \leq T$, or $\text{Max}(M, N) \leq T$, etc.), and the sub-blocks have a size of $C \times D$ when $M \times N > T$ (or $\text{Min}(M, N) > T$, or $\text{Max}(M, N) > T$, etc.), where $A \leq C$ and $B \leq D$. For example, if $M \times N \leq 256$, sub-blocks can be in a size of 4×4 . In some implementations, the sub-blocks have a size of 8×8 .

[0096] In some embodiments, whether to apply interweaved prediction can be determined based on the inter-prediction direction. The direction indicates whether the first or the second intermediate prediction block is predicted back in time or forward in time. For example, in some embodiments, the interweaved prediction may be applied for bi-prediction but not for uni-prediction.

[0097] As another example, when multiple-hypothesis is applied, the interweaved prediction may be applied for one prediction direction when there are more than one reference blocks. Multiple hypothesis can indicate that multiple video frames are used to make the prediction block. When multiple video frames are used to make the prediction block, the interweaved prediction can be applied to a prediction direction. The prediction direction can be a forward or a backward prediction direction. The forward prediction direction means that the multiple video

frames occur before the prediction block in a video sequence, while backward direction means that the multiple reference frames occur after the prediction block in the video sequence.

[0098] In some embodiments, how to apply interweaved prediction may also be determined based on the inter-prediction direction. In some embodiments, a bi-predicted block with sub-block based prediction is divided into sub-blocks with two different dividing patterns for two different reference lists. In bi-prediction, a first reference list and a second reference list can be obtained, where the first reference list and the second reference list represent frames forward and backward in time away from the predicted block. More specifically, the first reference list can include a first set of blocks that are in a first direction with respect to the prediction block in the video sequence. The first set of blocks can be used to create the prediction block. The second reference list can include a second set of blocks that are in a second direction with respect to the prediction block in video sequence. The second plurality of blocks can be used to create the prediction block. The first direction and the second direction can be opposite of each other, i.e. one direction can be forward, and the other direction can be backward in time away from the prediction block. The first reference list can be partitioned into the first set of sub-blocks according to the first dividing pattern. The second reference list can be partitioned into the second set of sub-blocks according to the second dividing pattern, where the first pattern and the second pattern are different.

[0099] For example, a bi-predicted block is divided into 4×8 sub-blocks as shown in FIG.

14D when predicted from reference list 0 (L0). The same block is divided into 8×4 sub-blocks as shown in FIG. 14C when predicted from reference list 1 (L1). The final prediction P is calculated as

$$P(x, y) = \frac{w^0(x, y) \times P^0(x, y) + w^1(x, y) \times P^1(x, y)}{w^0(x, y) + w^1(x, y)} \quad \text{Eq. (18)}$$

[00100] Here, P^0 and P^1 are predictions from L0 and L1, respectively. w^0 and w^1 are weighting values for L0 and L1, respectively. As shown in Eq. (16), the weighting values can be determined as: $w^0(x, y) + w^1(x, y) = 1 < N$ (wherein N is non-negative integer value). Because fewer sub-blocks are used for prediction in each direction (e.g., 4×8 sub-blocks as opposed to 8×8 sub-blocks), the computation requires less bandwidth as compared to the existing sub-block based methods. By using larger sub-blocks, the prediction results are also less susceptible to noise interference.

[00101] In some embodiments, a uni-predicted block with sub-block based prediction is

divided into sub-blocks with two or more different dividing patterns for the same reference list.

For example, the prediction for list L ($L=0$ or 1) P^L is calculated as

$$P^L(x, y) = \frac{\sum_{i=0}^{XL-1} w_i^L(x, y) \times P_i^L(x, y)}{\sum_{i=0}^{XL-1} w_i^L(x, y)} \quad \text{Eq. (19)}$$

[00102] Here XL is the number of dividing patterns for list L . $P_i^L(x, y)$ is the prediction

5 generated with the i^{th} dividing pattern and $w_i^L(x, y)$ is the weighting value of $P_i^L(x, y)$. For example, when XL is 2, two dividing patterns are applied for list L . In the first dividing pattern, the block is divided into 4×8 sub-blocks as shown in FIG. 14D. In the second dividing pattern, the block is divided into 8×4 sub-blocks as shown in FIG. 14C.

[00103] In some embodiments, a bi-predicted block with sub-block based prediction is

10 considered as a combination of two uni-predicted block from $L0$ and $L1$ respectively. The prediction from each list can be derived as described in the above example. The final prediction P can be calculated as

$$P(x, y) = \frac{a * \frac{\sum_{i=0}^{XL0-1} w_i^0(x, y) \times P_i^0(x, y)}{\sum_{i=0}^{XL0-1} w_i^0(x, y)} + b * \frac{\sum_{i=0}^{XL1-1} w_i^1(x, y) \times P_i^1(x, y)}{\sum_{i=0}^{XL1-1} w_i^1(x, y)}}{a + b} \quad \text{Eq. (20)}$$

[00104] Here parameters a and b are two additional weights applied to the two internal

15 prediction blocks. In this specific example, both a and b can be set to 1. Similar to the example above, because fewer sub-blocks are used for prediction in each direction (e.g., 4×8 sub-blocks as opposed to 8×8 sub-blocks), the bandwidth usage is better than or on par with the existing sub-block based methods. At the same time, the prediction results can be improved by using larger sub-blocks.

20 **[00105]** In some embodiments, a single non-uniform pattern can be used in each uni-predicted block. For example, for each list L (e.g., $L0$ or $L1$), the block is divided into a different pattern (e.g., as shown in FIG. 14E or FIG. 14F). The use of a smaller number of sub-blocks reduces the demand on bandwidth. The non-uniformity of the sub-blocks also increases robustness of the prediction results.

25 **[00106]** In some embodiments, for a multiple-hypothesis coded block, there can be more than one prediction blocks generated by different dividing patterns for each prediction direction (or reference picture list). Multiple prediction blocks can be used to generate the final prediction with additional weights applied. For example, the additional weights may be set to $1/M$ wherein M is the total number of generated prediction blocks.

[00107] In some embodiments, the encoder can determine whether and how to apply the interweaved prediction. The encoder then can transmit information corresponding to the determination to the decoder at a sequence level, a picture level, a view level, a slice level, a Coding Tree Unit (CTU) (also known as a Largest Coding Unit (LCU)) level, a CU level, a PU level, a Tree Unit (TU) level, tile level, tile group level or a region level (which may include multiple CUs/PUs/TUs/LCUs). The information can be signaled in a Sequence Parameter Set (SPS), a view parameter set (VPS), a Picture Parameter Set (PPS), a Slice Header (SH), picture header, sequence header, tile level, tile group level, a CTU/LCU, a CU, a PU, a TU, or a first block of a region.

[00108] In some implementations, the interweaved prediction applies to existing sub-block methods like the affine prediction, ATMVP, STMVP, FRUC, or BIO. In such cases, no additional signaling cost is needed. In some implementations, new sub-block merge candidates generated by the interweaved prediction can be inserted into a merge list, e.g., interweaved prediction + ATMVP, interweaved prediction + STMVP, interweaved prediction + FRUC etc.

[00109] In some embodiments, a flag may be signaled to indicate whether interweaved prediction is used or not. Signaling the flag can include coding the flag in the video information. In one example, the flag may be signaled to indicate whether interweaved prediction is used or not, if the current block is affine inter-coded. In another example, the flag may be signaled to indicate whether interweaved prediction is used or not, if the current block is affine merge-coded and applies uni-prediction. In a third example, the flag may be signaled to indicate whether interweaved prediction is used or not, if the current block is affine merge-coded.

[00110] In some embodiments, interweaved prediction may be always used if the current block is affine merge-coded and applies uni-prediction. In some embodiments, interweaved prediction may be always used if the current block is affine merge-coded.

[00111] In some embodiments, the flag to indicate whether interweaved prediction is used or not may be inherited without being signaled. In one example, the inheritance may be used if the current block is affine merge-coded. In another example, the flag may be inherited from the flag of the neighboring block where the affine model is inherited from. In a third example, the flag is inherited from a predefined neighboring block such as the left or above neighboring block. In a fourth example, the flag may be inherited from the first encountered affine-coded neighboring block. The flag may be inferred to be zero if no neighboring block is affine-coded. In other

words, if no neighboring block is affine-coded, no interweaved prediction is applied. In a fifth example, the flag may be only inherited when the current block applies uni-prediction. In a sixth example, the flag may be only inherited when the current block and the neighbouring block to be inherited from are in the same CTU. In a seventh example, the flag may be only inherited when the current block and the neighboring block to be inherited from are in the same CTU row. In an eighth example, the flag may not be inherited from the flag of the neighbouring block when the affine model is derived from a temporal neighboring block. In a ninth example, the flag may not be inherited from the flag of a neighboring block which is not located in the same LCU or LCU row or video data processing unit (such as 64x64, or 128x128). How to signal and/or derive the flag may depend on the block dimension of the current block and/or coded information. The coded information includes information encoded in the video.

[00112] In some embodiments, interweaved prediction is not applied if the reference picture is the current picture. In one example, the flag to indicate whether interweaved prediction is used or not is not signaled if the reference frame is the current frame containing the prediction block. The reference frame is used as a basis to predict the prediction block.

[00113] In some embodiments, the dividing patterns to be used by the current block can be derived based on information from spatial and/or temporal neighboring blocks. For example, instead of relying on the encoder to signal the relevant information, both encoder and decoder can adopt a set of predetermined rules to obtain dividing patterns based on temporal adjacency (e.g., previously used dividing patterns of the same block) or spatial adjacency (e.g., dividing patterns used by neighboring blocks).

[00114] In some embodiments, the weighting values w can be fixed. For example, all dividing patterns can be weighted equally: $w_i(x, y) = 1$. In some embodiments, the weighting values can be determined based on positions of blocks as well as the dividing patterns used. For example, $w_i(x, y)$ may be different for different (x, y) . In some embodiments, the weighting values may further depend on the sub-block prediction based coding techniques (e.g., affine, or ATMVP) and/or other coded information (e.g., skip or non-skip modes, and/or MV information).

[00115] In some embodiments, the encoder can determine the weighting values, and transmit the values to the decoder at sequence level, picture level, slice level, CTU/LCU level, CU level, PU level, or region level (which may include multiple CUs/PUs/TUs/LCUs). The weighting values can be signaled in a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a Slice

Header (SH), a CTU/LCU, a CU, a PU, or a first block of a region. In some embodiments, the weighting values can be derived from the weighting values of a spatial and/or temporal neighboring block.

[00116] It is noted that the interweaved prediction techniques disclosed herein can be applied in one, some, or all coding techniques of sub-block based prediction. For example, the interweaved prediction techniques can be applied to affine prediction, while other coding techniques of sub-block based prediction (e.g., ATMVP, STMVP, FRUC or BIO) do not use the interweaved prediction. As another example, all of affine, ATMVP, and STMVP apply the interweaved prediction techniques disclosed herein.

[00117] FIG. 15A is an example flowchart of a method 1500 for improving motion prediction in a video system in accordance with the disclosed technology. The method 1500 includes, at 1502, selecting a set of pixels from a video frame to form a block. The method 1500 includes, at 1504, partitioning the block into a first set of sub-blocks according to a first pattern. The method 1500 includes, at 1506, generating a first intermediate prediction block based on the first set of sub-blocks. The method 1500 includes, at 1508, partitioning the block into a second set of sub-blocks according to a second pattern. At least one sub-block in the second set has a different size than a sub-block in the first set. The method 1500 includes, at 1510, generating a second intermediate prediction block based on the second set of sub-blocks. The method 1500 also includes, at 1512, determining a prediction block based on the first intermediate prediction block and the second intermediate prediction block.

[00118] In some embodiments, the first intermediate prediction block or the second intermediate prediction block is generated using at least one of (1) an affine prediction method, (2) an alternative temporal motion vector prediction method, (3) a spatial-temporal motion vector prediction method, (4) a bi-directional optical flow method, or (5) a frame-rate up conversion method.

[00119] In some embodiments, the sub-blocks in the first or the second set have a rectangular shape. In some embodiments, the sub-blocks in the first set of sub-blocks have non-uniform shapes. In some embodiments, the sub-blocks in the second set of sub-blocks have non-uniform shapes.

[00120] In some embodiments, the method includes determining the first pattern or the second pattern based on a size of the block. In some embodiments, the method includes determining the

first pattern or the second pattern based on information from a second block that is temporally or spatially adjacent to the block.

[00121] In some embodiments, partitioning the block into the first set of sub-blocks is performed for a motion prediction of the block in a first direction. In some embodiments,

5 partitioning the block into the second set of sub-blocks is performed for a motion prediction of the block in a second direction.

[00122] In some embodiments, partitioning the block into the first set of sub-blocks and partitioning the block into the second set of sub-blocks are performed for a motion prediction of the block in a first direction. In some embodiments, the method further includes performing a

10 motion prediction of the block in a second direction by partitioning the block into a third set of sub-blocks according to a third pattern, generating a third intermediate prediction block based on the third set of sub-blocks, partitioning the block into a fourth set of sub-blocks according to a fourth pattern, wherein at least one sub-block in the fourth set has a different size than a sub-block in the third set, generating a fourth intermediate prediction block based on the fourth set of

15 sub-blocks, determining a second prediction block based on the third intermediate prediction block and the fourth intermediate prediction block, and determining a third prediction block based on the prediction block and the second prediction block.

[00123] In some embodiments, the method includes transmitting, to a coding device in the block-based motion prediction video system, information of the first pattern and the second

20 pattern for partitioning the block. In some embodiments, transmitting the information of the first pattern and the second pattern is performed at one of: (1) a sequence level, (2) a picture level, (3) a view level, (4) a slice level, (5) a Coding Tree Unit, (6) a Largest Coding Unit level, (7) a Coding Unit level, (8) a Prediction Unit level, (10) a Tree Unit level, or (11) a region level.

[00124] In some embodiments, determining the prediction result includes applying a first set
25 of weights to the first intermediate prediction block to obtain a first weighted prediction block, applying a second set weights to the second intermediate prediction block to obtain a second weighted prediction block, and computing a weighted sum of the first weighted prediction block and the second weighted prediction block to obtain the prediction block.

[00125] In some embodiments, the first set of weights or the second set of weights includes
30 fixed- weight values. In some embodiments, the first set of weights or the second set of weights is determined based on information from another block that is temporally or spatially adjacent to

the block. In some embodiments, the first set of weights or the second set of weights is determined using a coding algorithm used for generating the first prediction block or the second prediction block. In some implementations, at least one value in the first set of weights is different than another value in the first set of weights. In some implementations, at least one value in the second set of weights is different than another value in the second set of weights. In some implementations, a sum of the weights is equal to a power of two.

[00126] In some embodiments, the method includes transmitting the weights to a coding device in the block-based motion prediction video system. In some embodiments, transmitting the weights is performed at one of: (1) a sequence level, (2) a picture level, (3) a view level, (4) a slice level, (5) a Coding Tree Unit, (6) a Largest Coding Unit level, (7) a Coding Unit level, (8) a Prediction Unit level, (10) a Tree Unit level, or (11) a region level.

[00127] FIG. 15B is an example flowchart of a method 1550 for improving block-based motion prediction in a video system in accordance with the disclosed technology. The method 1550 includes, at 1552, selecting a set of pixels from a video frame to form a block. The method 1550 includes, at 1554, dividing the block into multiple sub-blocks based on a size of the block or information from another block that is spatially or temporally adjacent to the block. At least one sub-block of the multiple sub-blocks has a different size than other sub-blocks. The method 1550 also includes, at 1556, generating motion vector predictions by applying a coding algorithm to the multiple sub-blocks. In some embodiments, the coding algorithm includes at least one of (1) an affine prediction method, (2) an alternative temporal motion vector prediction method, (3) a spatial-temporal motion vector prediction method, (4) a bi-directional optical flow method, or (5) a frame-rate up conversion method.

[00128] As further described in the present document, the encoding process may refrain from checking affine mode for blocks that are split from a parent block, where the parent block itself is encoded with a mode different from affine mode.

[00129] Table 1 illustrates example performance results using conventional 2x2 Affine prediction for a Random Access (RA) configuration.

Table 1 Example Test Results from 2x2 Affine Prediction

	Y	U	V	EncT	DecT
Class A1	-0.11%	-0.18%	-0.09%	139%	111%
Class A2	-0.9%	-0.85%	-0.68%	142%	125%

Class B	-0.58%	-0.51%	-0.67%	136%	114%
Class C	-0.26%	-0.24%	-0.24%	129%	108%
Class D	-0.54%	-0.52%	-0.53%	130%	118%
Class F	-0.89%	-1.02%	-0.97%	125%	108%
Overall	-0.47%	-0.44%	-0.44%	136%	114%

[00130] Table 2 illustrates example performance results obtained by applying interweaved prediction on uni-prediction in accordance with an embodiment of the present technology. Table 3 illustrates example performance results obtained by applying interweaved prediction on bi-prediction in accordance with an embodiment of the present technology.

Table 2 Example Test Results from Interleaved Prediction in Uni-prediction

	Y	U	V	EncT	DecT
Class A1	-0.05%	-0.14%	-0.02%	101%	100%
Class A2	-0.55%	-0.17%	-0.11%	102%	101%
Class B	-0.33%	-0.17%	-0.20%	101%	101%
Class C	-0.15%	-0.16%	-0.04%	100%	100%
Class D	-0.21%	-0.09%	-0.02%	106%	106%
Class F	-0.39%	-0.40%	-0.39%	102%	102%
Overall	-0.27%	-0.16%	-0.11%	101%	101%

Table 3 Example Test Results from Interleaved Prediction in Bi-prediction

	Y	U	V	EncT	DecT
Class A1	-0.09%	-0.18%	-0.12%	103%	102%
Class A2	-0.74%	-0.40%	-0.28%	104%	104%
Class B	-0.37%	-0.39%	-0.35%	103%	102%
Class C	-0.22%	-0.19%	-0.13%	102%	102%
Class D	-0.42%	-0.28%	-0.32%	103%	102%
Class F	-0.60%	-0.64%	-0.62%	102%	102%

Overall	-0.38%	-0.30%	-0.23%	103%	102%
---------	--------	--------	--------	------	------

[00131] As shown in Table 2 and Table 3, interweaved prediction achieves major coding gain with much lower complexity, compared to conventional 2x2 affine prediction based coding. In particular, interweaved prediction applied in bi-prediction achieves a coding gain of 0.38% as compared to 2x2 Affine method (0.47%). The encoding time and decoding time are 103% and 102% respectively as compared to 136% and 114% in the 2x2 Affine method.

[00132] FIG. 16 is a block diagram illustrating an example of the architecture for a computer system or other control device 1600 that can be utilized to implement various portions of the presently disclosed technology. In FIG. 16, the computer system 1600 includes one or more processors 1605 and memory 1610 connected via an interconnect 1625. The interconnect 1625 may represent any one or more separate physical buses, point to point connections, or both, connected by appropriate bridges, adapters, or controllers. The interconnect 1625, therefore, may include, for example, a system bus, a Peripheral Component Interconnect (PCI) bus, a HyperTransport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), IIC (I2C) bus, or an Institute of Electrical and Electronics Engineers (IEEE) standard 674 bus, sometimes referred to as "Firewire."

[00133] The processor(s) 1605 may include central processing units (CPUs) to control the overall operation of, for example, the host computer. In certain embodiments, the processor(s) 1605 accomplish this by executing software or firmware stored in memory 1610. The processor(s) 1605 may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic devices (PLDs), or the like, or a combination of such devices.

[00134] The memory 1610 can be or include the main memory of the computer system. The memory 1610 represents any suitable form of random access memory (RAM), read-only memory (ROM), flash memory, or the like, or a combination of such devices. In use, the memory 1610 may contain, among other things, a set of machine instructions which, when executed by processor 1605, causes the processor 1605 to perform operations to implement embodiments of the presently disclosed technology.

[00135] Also connected to the processor(s) 1605 through the interconnect 1625 is a (optional)

network adapter 1615. The network adapter 1615 provides the computer system 1600 with the ability to communicate with remote devices, such as the storage clients, and/or other storage servers, and may be, for example, an Ethernet adapter or Fiber Channel adapter.

[00136] FIG. 17 shows a block diagram of an example embodiment of a mobile device 1700

that can be utilized to implement various portions of the presently disclosed technology. The mobile device 1700 can be a laptop, a smartphone, a tablet, a camcorder, or other types of

devices that are capable of processing videos. The mobile device 1700 includes a processor or controller 1701 to process data, and memory 1702 in communication with the processor 1701 to store and/or buffer data. For example, the processor 1701 can include a central processing unit

(CPU) or a microcontroller unit (MCU). In some implementations, the processor 1701 can

include a field-programmable gate-array (FPGA). In some implementations, the mobile device 1700 includes or is in communication with a graphics processing unit (GPU), video processing unit (VPU) and/or wireless communications unit for various visual and/or communications data processing functions of the smartphone device. For example, the memory 1702 can include and

store processor-executable code, which when executed by the processor 1701, configures the mobile device 1700 to perform various operations, e.g., such as receiving information,

commands, and/or data, processing information and data, and transmitting or providing processed information/data to another device, such as an actuator or external display. To support

various functions of the mobile device 1700, the memory 1702 can store information and data,

such as instructions, software, values, images, and other data processed or referenced by the

processor 1701. For example, various types of Random Access Memory (RAM) devices, Read Only Memory (ROM) devices, Flash Memory devices, and other suitable storage media can be used to implement storage functions of the memory 1702. In some implementations, the mobile

device 1700 includes an input/output (I/O) unit 1703 to interface the processor 1701 and/or

memory 1702 to other modules, units or devices. For example, the I/O unit 1703 can interface the processor 1701 and memory 1702 with to utilize various types of wireless interfaces

compatible with typical data communication standards, e.g., such as between the one or more

computers in the cloud and the user device. In some implementations, the mobile device 1700

can interface with other devices using a wired connection via the I/O unit 1703. The mobile

device 1700 can also interface with other external interfaces, such as data storage, and/or visual

or audio display devices 1704, to retrieve and transfer data and information that can be processed

by the processor, stored in the memory, or exhibited on an output unit of a display device 1704 or an external device. For example, the display device 1704 can display a video frame modified based on the MVPs (e.g., a video frame that includes the prediction block 1305 as shown in FIG. 13) in accordance with the disclosed technology.

5 [00137] In some embodiments, a video decoder apparatus may implement a method of video decoding in which the improved block-based motion prediction as described herein is used for video decoding. The method may include forming a block of video using a set of pixels from a video frame. The block may be partitioned into a first set of sub-blocks according to a first pattern. A first intermediate prediction block may correspond to the first set of sub-blocks. The
10 block may include a second set of sub-blocks according to a second pattern. At least one sub-block in the second set has a different size than a sub-block in the first set. The method may further determine a prediction block based on the first intermediate prediction block and a second intermediate prediction block that is generated from the second set of sub-blocks. Other features of this method may be similar to the above-described method 1500.

15 [00138] In some embodiments, a decoder-side method of video decoding may use block-based motion prediction for improving video quality by using blocks of a video frame for prediction, where a block corresponds to a set of pixel blocks. The block may be divided into multiple sub-blocks based on a size of the block or information from another block that is spatially or temporally adjacent to the block, wherein at least one sub-block of the multiple sub-blocks has a
20 different size than other sub-blocks. The decoder may use motion vector predictions that are generated by applying a coding algorithm to the multiple sub-blocks. Other features of this method are described with respect to FIG. 15B and the corresponding description.

[00139] In some embodiments, the video decoding methods may be implemented using a decoding apparatus that is implemented on a hardware platform as described with respect to FIG.
25 16 and FIG. 17.

[00140] FIG. 8 is a flowchart for an example method 800 of video coding or decoding. The method 800 includes determining (802), due to a block meeting a condition, to apply interweaved prediction to the block. The method 800 includes determining (804) a prediction block based on a first intermediate prediction block and a second intermediate prediction block.
30 The method 800 includes generating (804) an encoded or a decoded representation of the block using the prediction block. For example, a video encoder or transcoder may perform encoding at

804 and a video decoder may perform generation of a decoded representation at 804. The first intermediate prediction block is generated from the first set of sub-blocks obtained by partitioning the block according to a first pattern and the second intermediate prediction block is generated from the second set of sub-blocks obtained by partitioning the block according to a second pattern and at least one sub-block in the second set has a different size than a sub-block in the first set.

[00141] In some embodiments, the condition met by the block is that the block being coded using bi-prediction coding. In some embodiments, the condition met by the block is that the block is predicted using a multiple-hypothesis prediction and wherein the interweaved prediction is applied in a prediction direction where multiple reference blocks are available. In some embodiments, the partitioning the block into the first set of sub-blocks and the partitioning the block into the second set of sub-blocks are performed for a motion prediction of the block in a first direction. In some embodiments, the coded representation, may include information of the first pattern and the second pattern.

[00142] In some embodiments, the condition includes the block being coded using bi-prediction but not to a uni-prediction, wherein the bi-prediction is based on a preceding and a subsequent video frame, and the uni-prediction is based on only the preceding or the subsequent video frame. In some embodiments, the condition is that the block is bi-predicted, and wherein the first intermediate prediction block is generated from the first set of sub-blocks using a first reference list for the block and the second intermediate prediction block is generated from the second set of sub-blocks using a second reference list for the block. In some embodiments, the condition is that the block is uni-predicted and wherein the first intermediate prediction block is generated from the first set of sub-blocks using a reference list L0 or L1 for the block and the second intermediate prediction block is generated from the second set of sub-blocks using the reference list L0 or L1 for the block. In some embodiments, the condition is that the block is bi-predicted, and wherein the first intermediate prediction block is generated from the first set of sub-blocks using a first reference list for the block and the second intermediate prediction block is generated from the second set of sub-blocks using a second reference list for the block. The method may further include generating, using the first reference list, one or more third intermediate prediction blocks from one or more third sets of sub-blocks of the block and generating, using the second reference list, one or more fourth intermediate prediction blocks

from one or more fourth set of sub-blocks of the block, wherein the prediction block is determined based on the one or more third intermediate prediction blocks and/or the one or more fourth intermediate prediction blocks.

[00143] In some embodiments, the condition is that the block is a multiple-hypothesis coded block. The method may further include generating, using a reference list used for the block, one or more additional intermediate prediction blocks from one or more additional sets of sub-blocks of the block, wherein the prediction block is determined based on the one or more additional intermediate prediction blocks.

[00144] Additional features and variations of the method 800 may be similar to as described with reference to FIGS. 15A and 15B.

[00145] FIG. 9 shows a functional block diagram of an example apparatus 1900 that implements the interweaved prediction techniques disclosed in the present document. For example, the apparatus 1900 may be a video encoder or transcoder that receives video 1902. The received video 1902 may be in the form of compressed video or uncompressed video. The video 1902 may be received through a network interface or from a storage device. The video 1902 (either in uncompressed form or in compressed form) may correspond to video frames of a certain size. The apparatus 1900 may perform a pre-processing 1904 operation on the video 1902. The pre-processing 1904 may be optional and may include things such as decryption, color space conversion, filtering for quality enhancement, etc. The encoder 1906 may convert the video 1902 into a coded representation that may be optionally post-processed through a post-processing block 1910 to produce the output video. The encoder 1906 may perform, for example, interweaved prediction on blocks of video 1902. A block may represent a video region of any size but may often be selected to have a fixed number of horizontal and vertical dimensions in number of pixels (e.g., 128x128 or 16x16, etc.). In some cases, the block may represent a coding unit. The optional post-processing block may include filtering, encryption, packetization, and so on. The output video 1910 may be stored on a memory device or may be transmitted via a network interface.

[00146] From the foregoing, it will be appreciated that specific embodiments of the presently disclosed technology have been described herein for purposes of illustration, but that various modifications may be made without deviating from the scope of the invention. Accordingly, the presently disclosed technology is not limited except as by the appended claims.

[00147] The disclosed and other embodiments, modules and the functional operations described in this document can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this document and their structural equivalents, or in combinations of one or more of them. The disclosed and other
5 embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a
10 combination of one or more them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database
15 management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[00148] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or
20 interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a
25 markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00149] The processes and logic flows described in this document can be performed by one or
30 more programmable processors executing one or more computer programs to perform functions

by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[00150] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[00151] While this patent document contains many specifics, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[00152] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described

in this patent document should not be understood as requiring such separation in all embodiments.

[00153] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this

5 patent document.

CLAIMS

What is claimed is:

1. A method for processing video blocks with interweaved prediction, the method
5 comprising:
determining, due to a block meeting one or more conditions, a prediction block for the
block based on a first intermediate prediction block and a second intermediate prediction block;
and
generating an encoded or a decoded representation of the block using the prediction
10 block;
wherein the first intermediate prediction block is generated from the first set of sub-
blocks obtained by partitioning the block according to a first dividing pattern and the second
intermediate prediction block is generated from the second set of sub-blocks obtained by
partitioning the block according to a second dividing pattern,
15 wherein the first dividing pattern is different from the second dividing pattern.
2. The method of claim 1, wherein the condition is that the block is uni-predicted and
wherein the first intermediate prediction block is generated from the first set of sub-blocks and
the second intermediate prediction block is generated from the second set of sub-blocks, from the
20 same reference picture.
3. The method of claim 1, wherein the condition met by the block is that the block being
coded using bi-prediction coding.
- 25 4. The method of claim 1, wherein the condition includes the block being coded using bi-
prediction but not to a uni-prediction, wherein the bi-prediction is based on both reference list L0
and reference list L1, and the uni-prediction is based on only reference list L0 or reference list
L1.
- 30 5. The method of claim 1, wherein the condition is that the block is bi-predicted, and
wherein the first intermediate prediction block is generated from the first set of sub-blocks using

a first reference list for the block and the second intermediate prediction block is generated from the second set of sub-blocks using a second reference list for the block.

6. The method of claim 1, wherein the condition met by the block is that the block is predicted using a multiple-hypothesis prediction and wherein the interweaved prediction is applied in a prediction direction where multiple reference blocks are available.

7. The method of claim 1, wherein the condition is that the block is bi-predicted, and wherein the first intermediate prediction block is generated from the first set of sub-blocks using a first reference list for the block and the second intermediate prediction block is generated from the second set of sub-blocks using a second reference list for the block, the method further including:

generating, using the first reference list, one or more third intermediate prediction blocks from one or more third sets of sub-blocks of the block;

generating, using the second reference list, one or more fourth intermediate prediction blocks from one or more fourth set of sub-blocks of the block;

wherein the prediction block is determined based on the one or more third intermediate prediction blocks and/or the one or more fourth intermediate prediction blocks.

8. The method of claim 1, wherein the condition is that the block is a multiple-hypothesis coded block, the method further including:

generating, using a reference list used for the block, one or more additional intermediate prediction blocks from one or more additional sets of sub-blocks of the block; and

wherein the prediction block is determined based on the one or more additional intermediate prediction blocks.

9. The method of claim 8, wherein the prediction block is determined to be an equally-weighted sum of the first intermediate prediction block, the second intermediate prediction block and the one or more additional intermediate prediction blocks.

10. The method of claim 1, wherein the partitioning the block into the first set of sub-blocks and the partitioning the block into the second set of sub-blocks are performed for a motion prediction of the block in a first direction.
- 5 11. The method of any of claims 1 to 10, further comprising:
including, in the coded representation, information of the first dividing pattern and the second dividing pattern.
- 10 12. The method of claim 11, wherein the information of the first dividing pattern and the second dividing pattern is included at one of: (1) a sequence level, (2) a picture level, (3) a view level, (4) a slice level, (5) a Coding Tree Unit, (6) a Largest Coding Unit level, (7) a Coding Unit level, (8) a Prediction Unit level, (10) a Tree Unit level, or (11) a region level.
- 15 13. The method of any of claims 1-12, wherein the coded representation includes information about the interweaved prediction at (1) a sequence level, (1) a picture level, (3) a view level, (4) a slice level, (5) a Coding Tree Unit (CTU), (6) a largest coding unit (LCU) level, (7) a coding unit (CU) level, (8) a prediction unit (PU) level, (9) a tree unit (TU) level, (10) tile level, (11) tile group level or (12) a region level which may contain multiple CUs/PUs/TUs/LCUs.
- 20 14. The method of any of claims 1 to 13 wherein the coded representation includes information regarding whether and how to apply the interweaved prediction in a Sequence Parameter Set (SPS), a view parameter set (VPS), a Picture Parameter Set (PPS), a Slice Header (SH), a picture header, a sequence header, a tile level, a tile group level, or an initial block of a region, wherein the information includes a flag that is included selectively based on the condition of the block.
- 25 15. The method of claim 14, wherein the flag is omitted when the block is coded using a sub-block prediction method.
16. The method of claim 15, wherein the sub-block prediction method is alternative temporal motion vector predictor method.

17. The method of claim 15, wherein the sub-block prediction method is affine prediction method.
18. The method of claim 15, wherein the sub-block prediction method is a frame rate up conversion method.
- 5 19. The method of claim 15, wherein the sub-block prediction method is a bidirectional optical flow method.
20. The method of claim 15, wherein the sub-block prediction method is a spatio temporal motion vector prediction method.
- 10 21. The method of claim 14, comprising signaling a flag to indicate whether the interweaved prediction is used or not, if the block is affine-coded.
22. The method of claim 14, wherein a flag to indicate whether the interweaved prediction is used or not, without signaling the flag.
- 15 23. The method of any of claims 1 to 22, a flag indicating whether or not interweaved prediction is used for the block is inherited from previously coded information in the coded representation.
- 20 24. The method of claim 23, comprising inheriting the flag from a neighboring block from which an affine model is inherited.
- 25 25. The method of claim 24, comprising inheriting the flag of a predefined neighboring block comprising a left neighboring block or an above neighbouring block.
26. The method of claim 23, comprising inheriting the flag from an initially encountered affine-coded neighbouring block.
27. The method of claim 23, comprising inferring that there is no interweaved prediction when no neighbouring block is affine-coded.

28. The method of claim 23, comprising inheriting the flag when the block applies uni-prediction.

5 29. The method of claim 23, comprising inheriting the flag when the block and a neighbouring block from which the flag is inherited from are in the same CTU.

30. The method of claim 23, comprising inheriting the flag when the current block and a neighbouring block to be inherited from are in the same CTU row.

10

31. The method any of claims 1-30, wherein the condition includes the width and height of the block.

15 32. The method of claim 1, wherein the condition includes that the block is coded without using another block of the video frame.

33. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of claims 1-32.

20 34. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of claims 1 to 32.

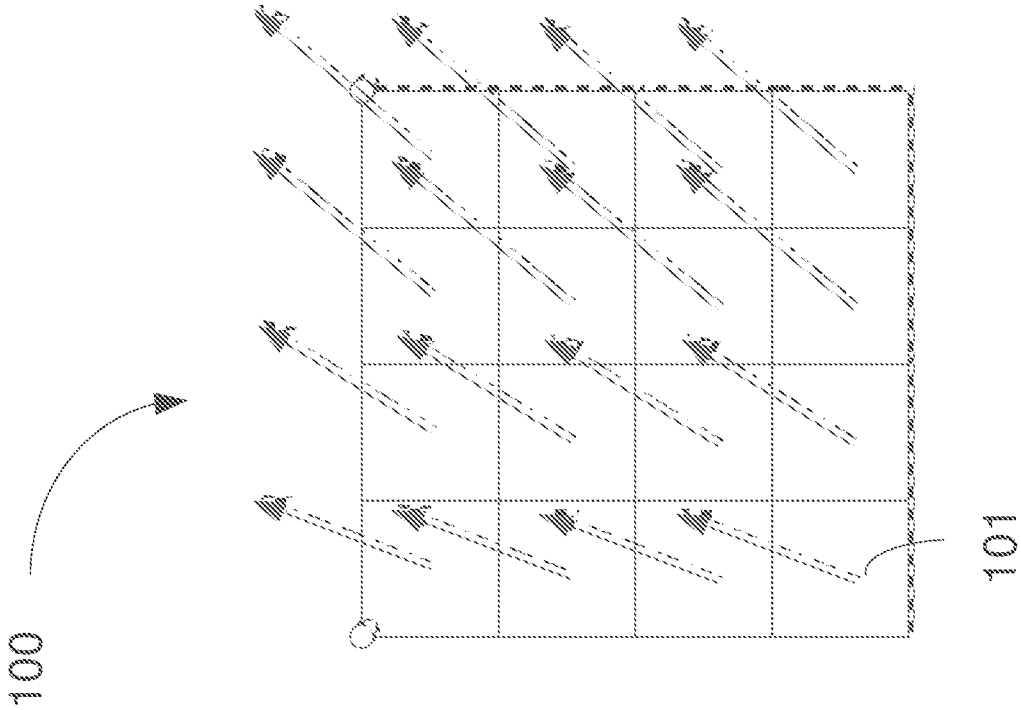


FIG. 1

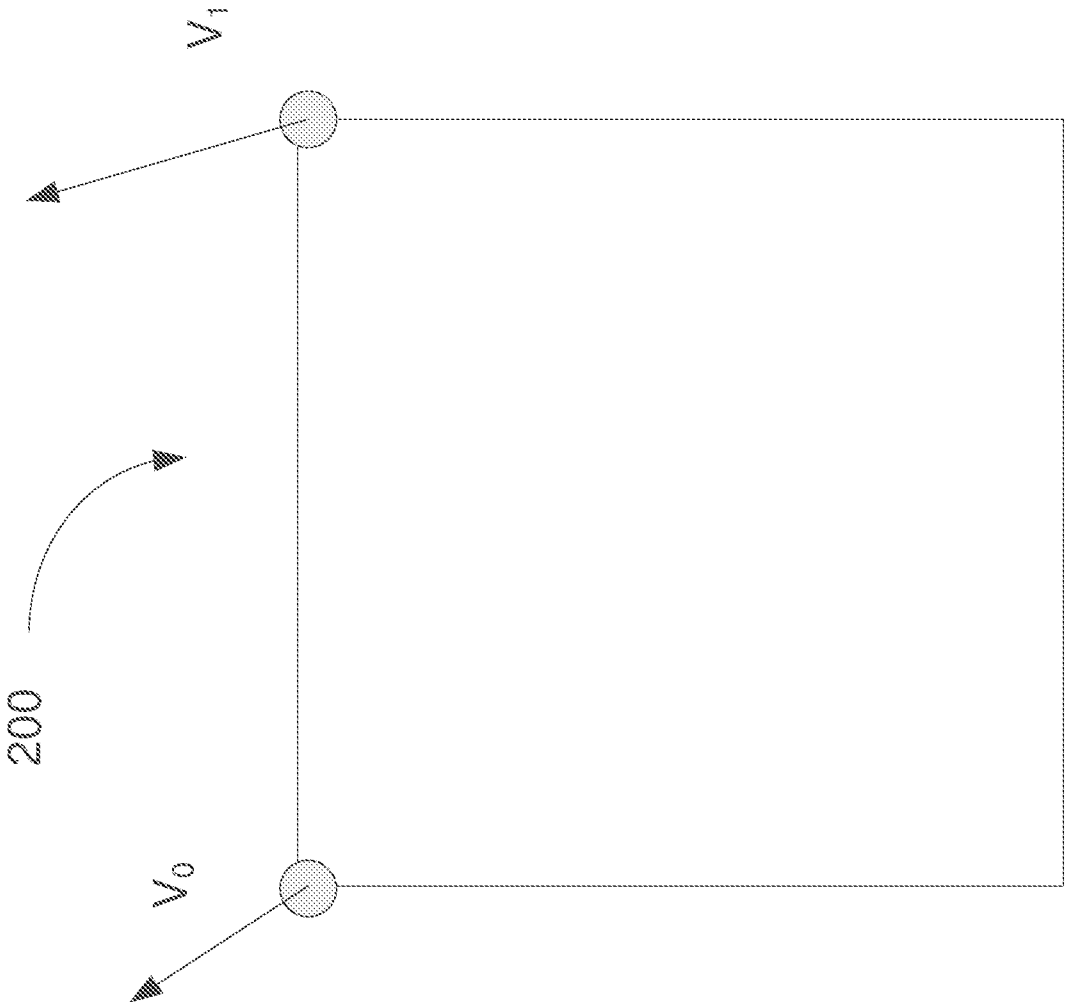


FIG. 2

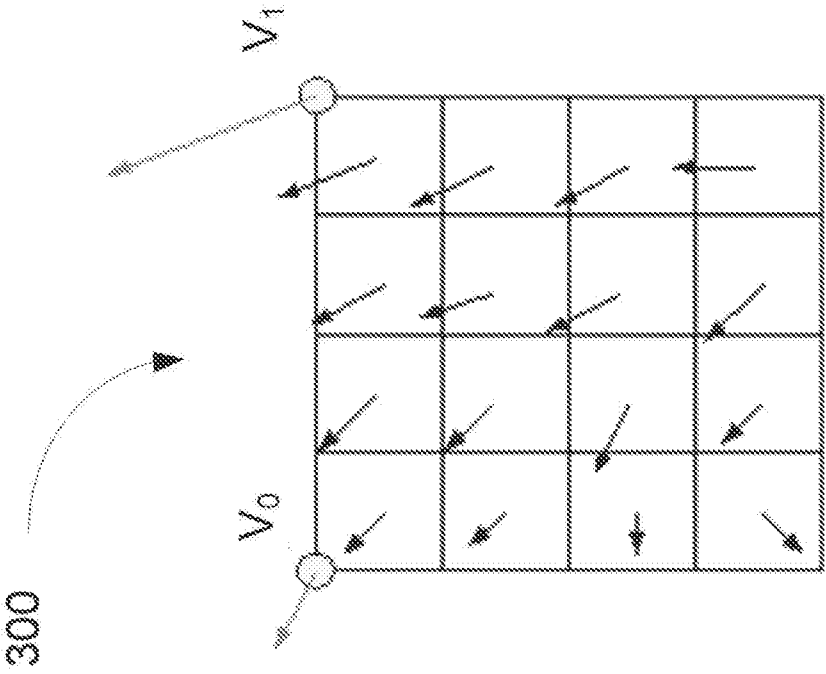


FIG. 3

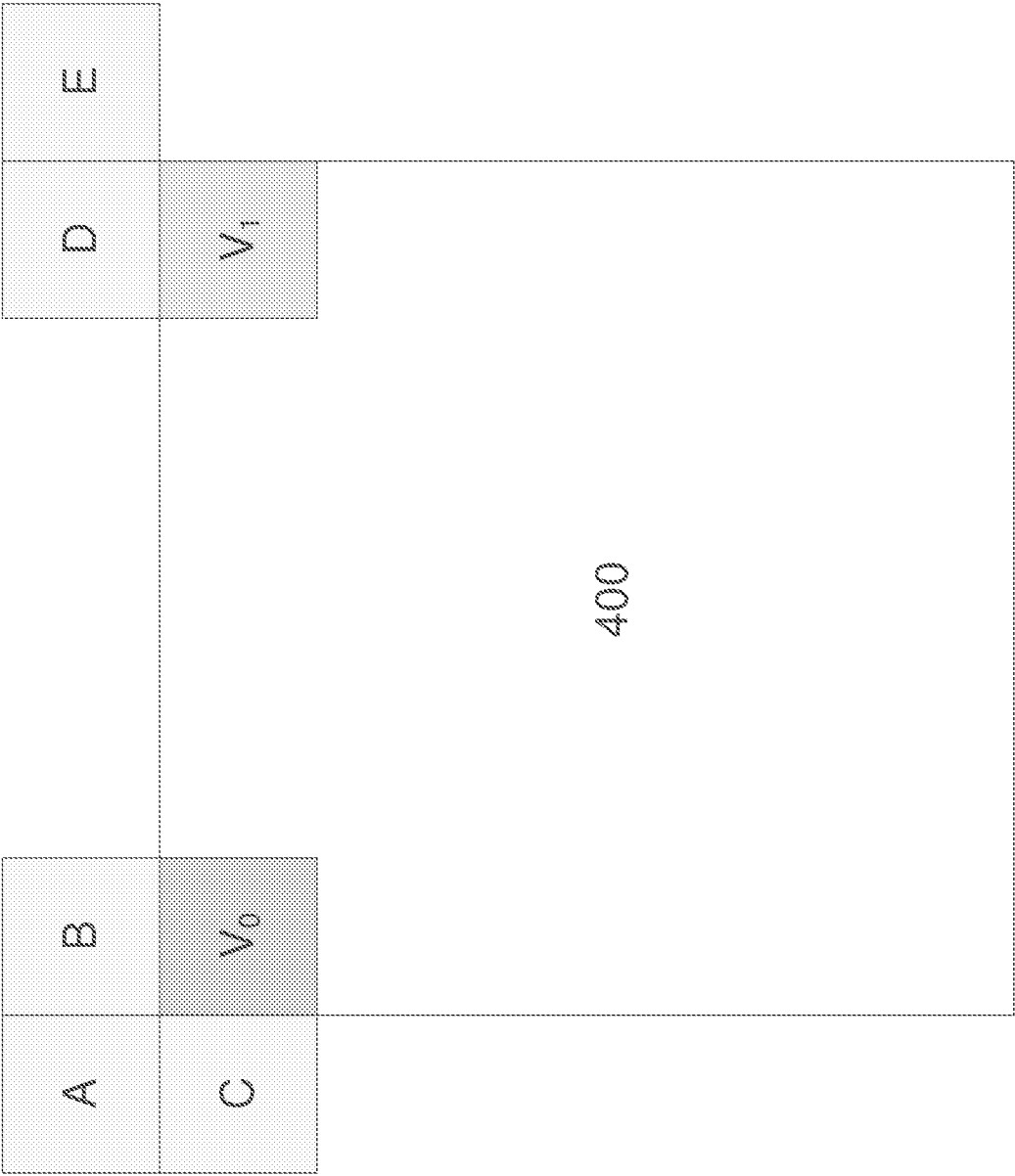


FIG. 4

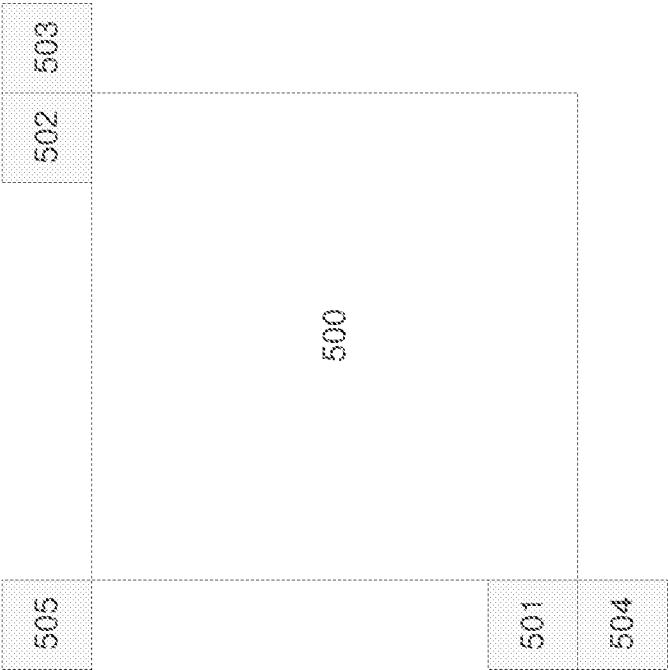


FIG. 5A

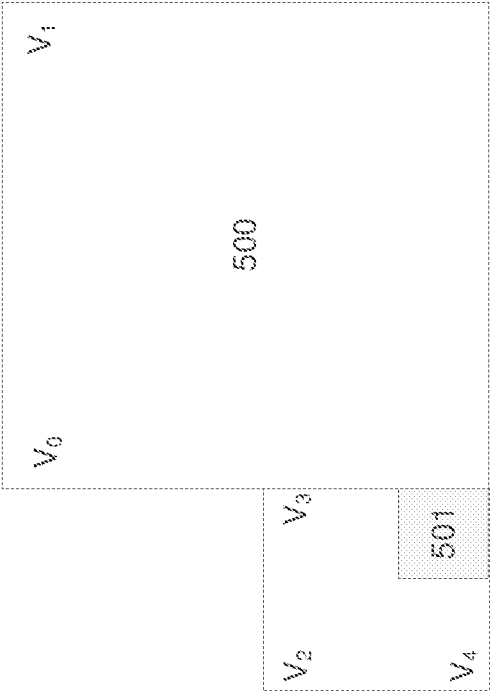


FIG. 5B

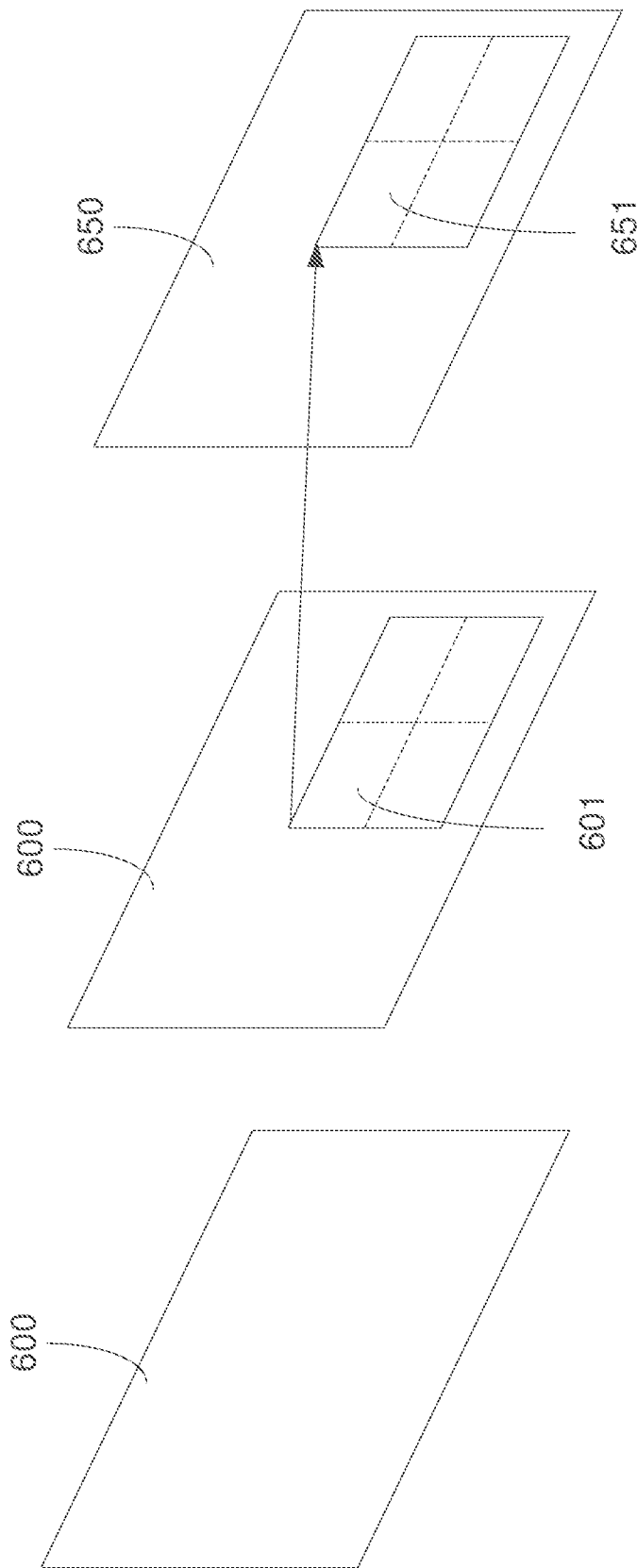


FIG. 6



FIG. 7

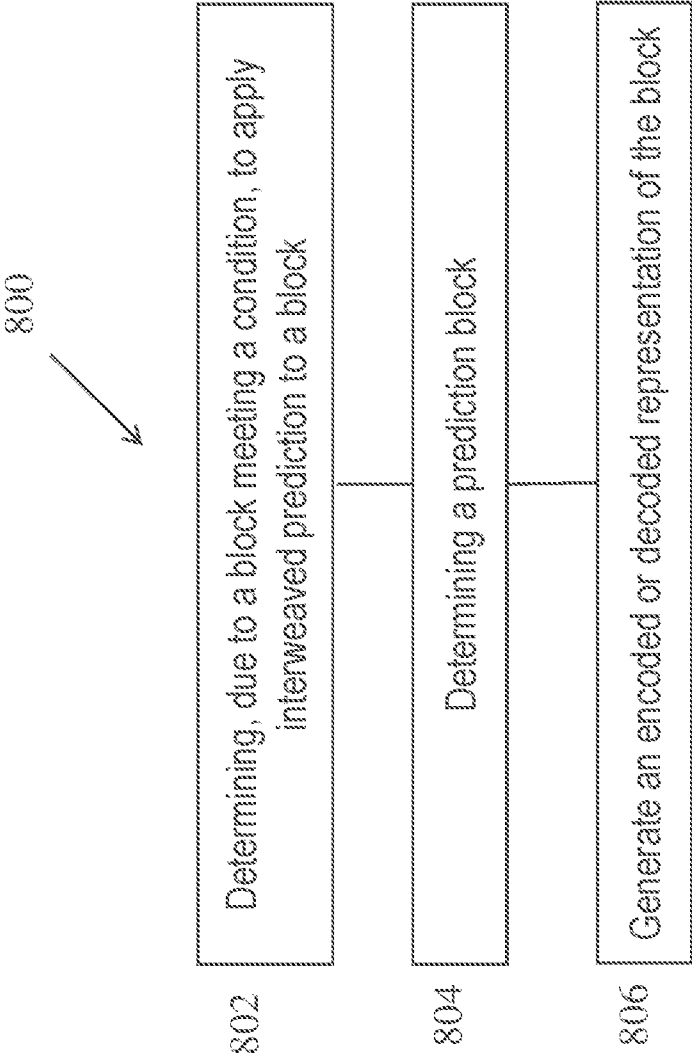


FIG. 8

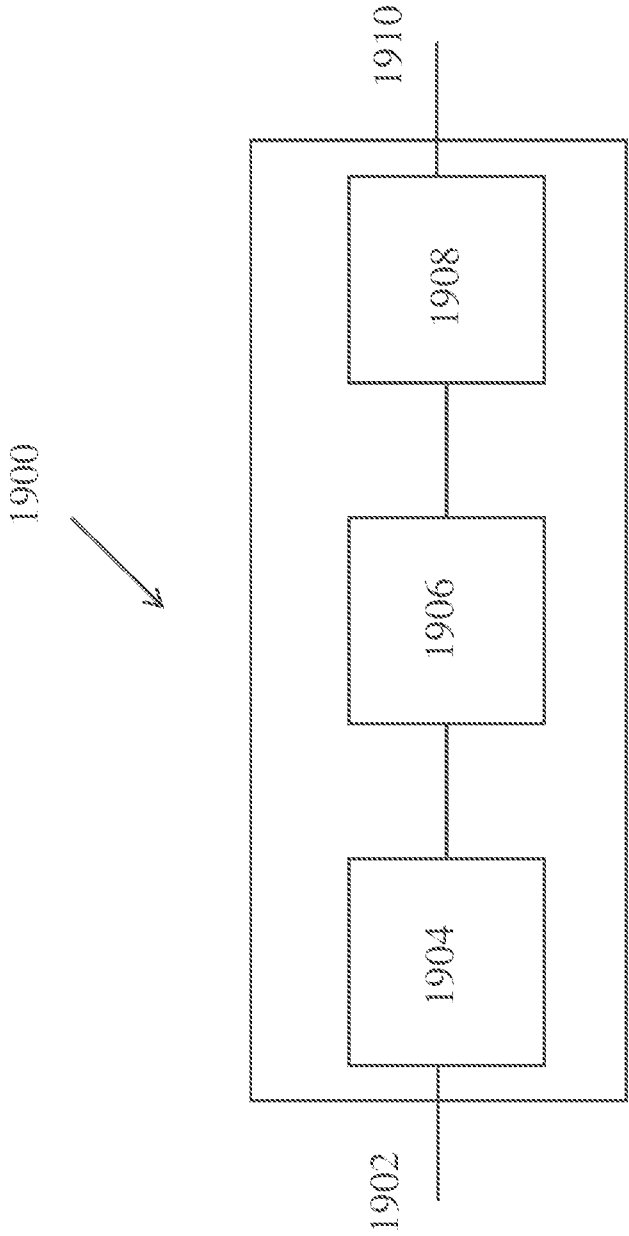


FIG. 9

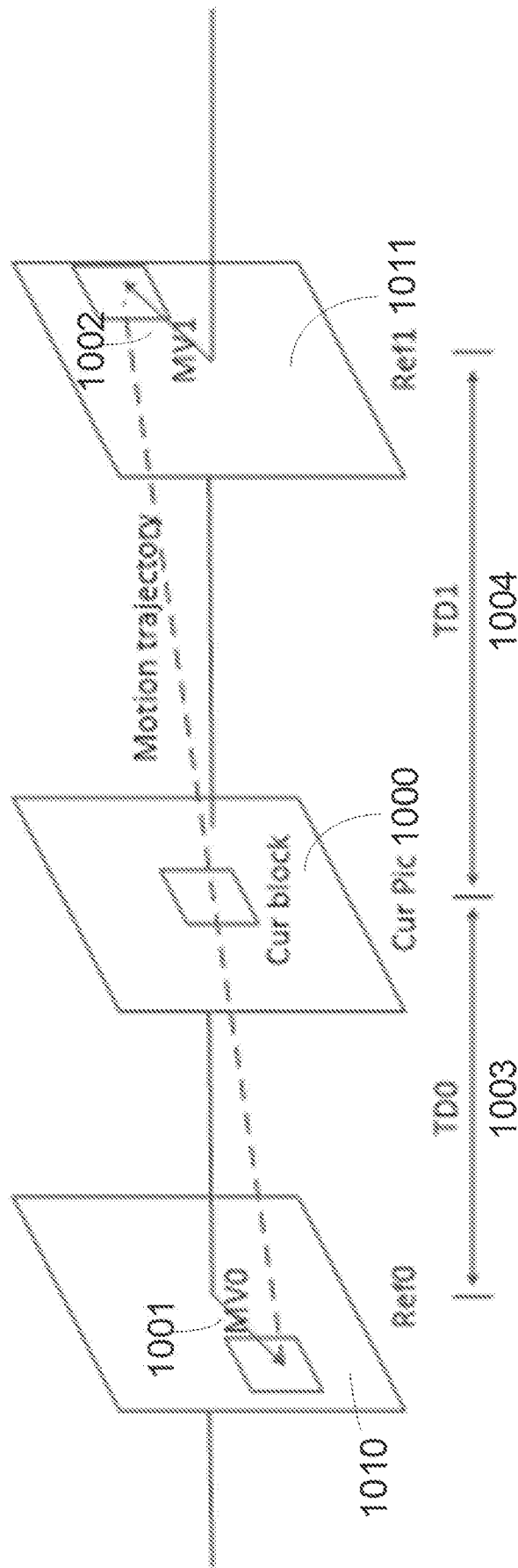


FIG. 10

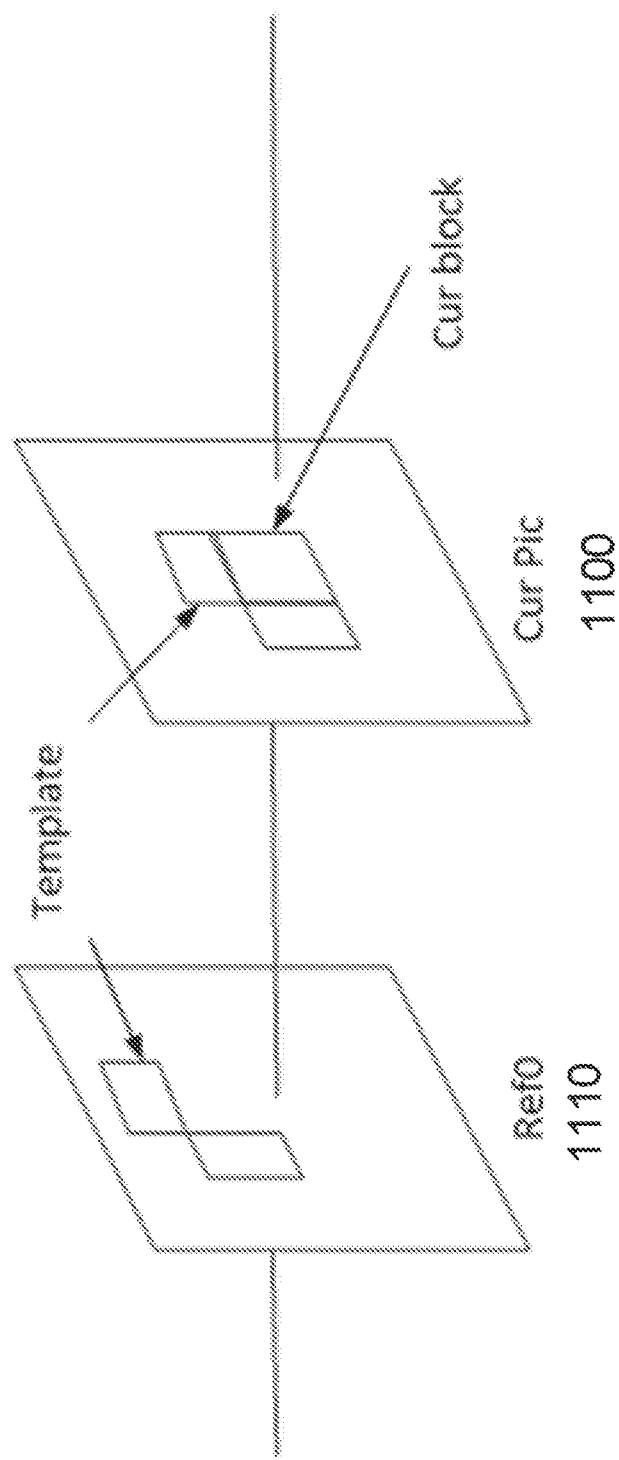


FIG. 11

1200

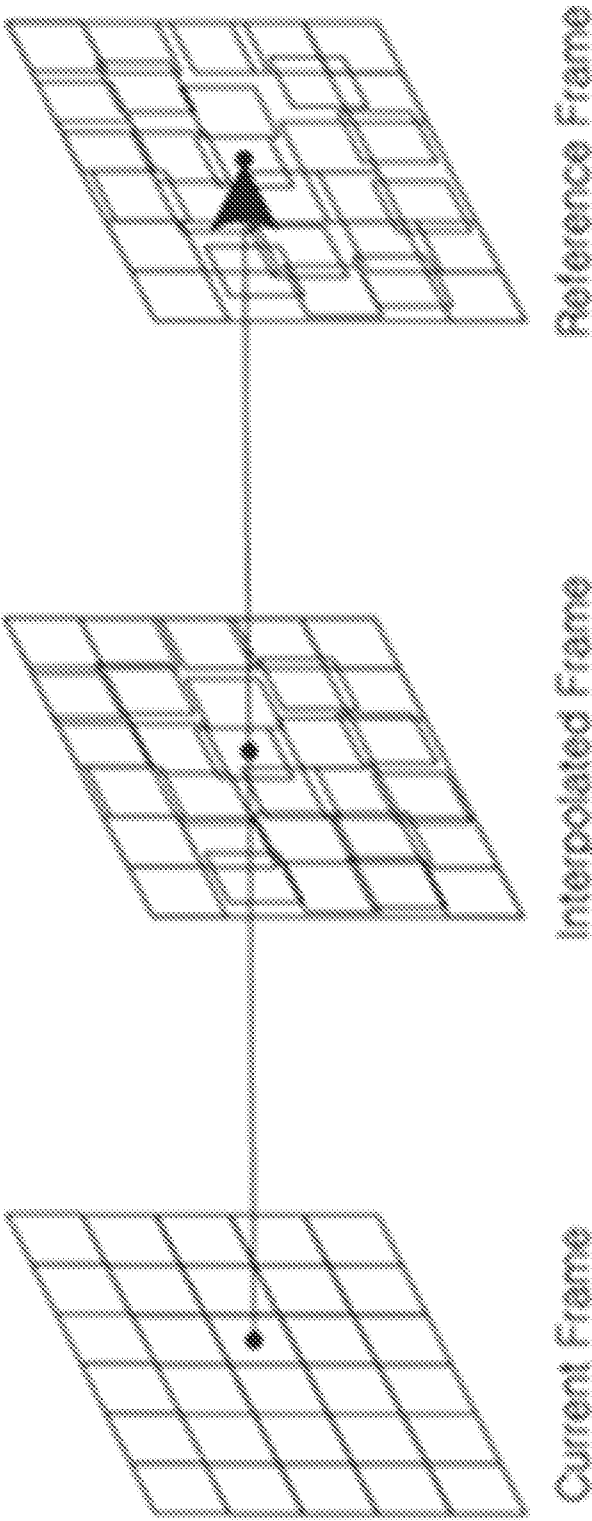


FIG. 12

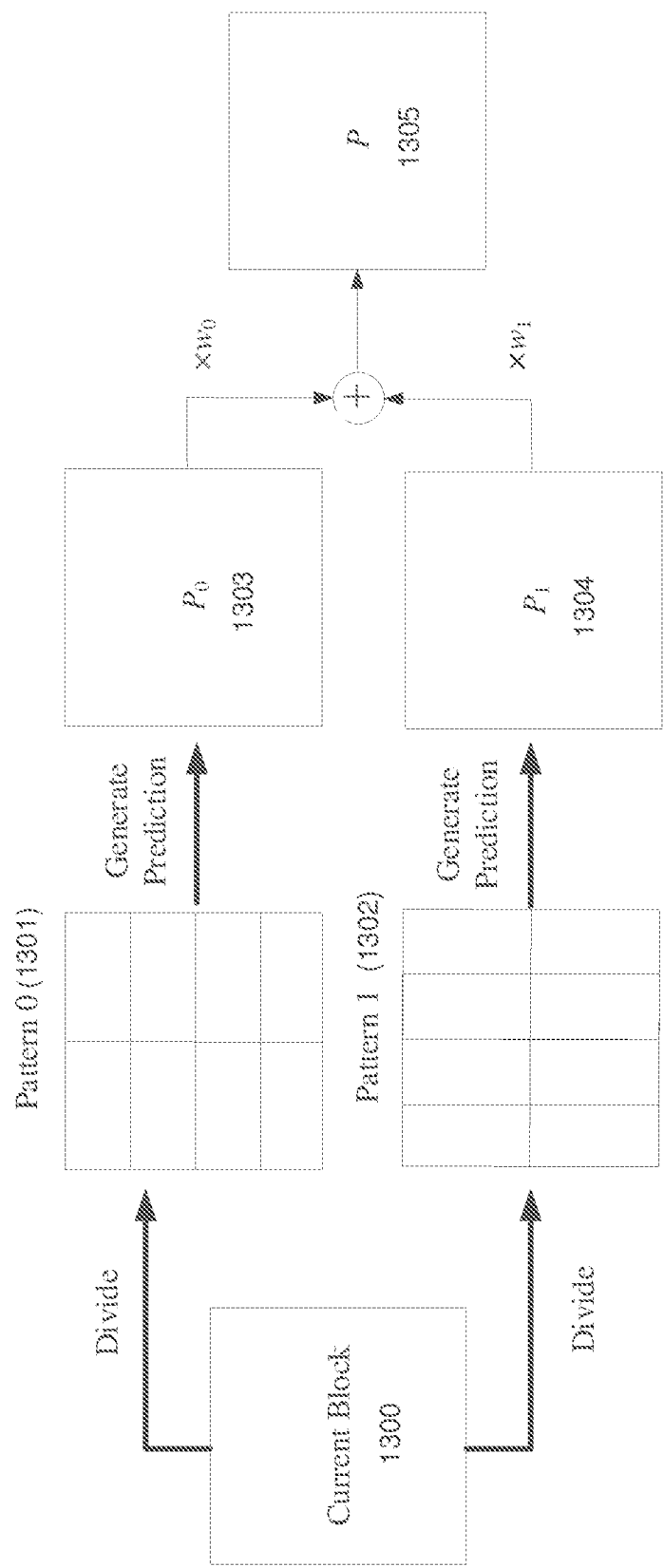
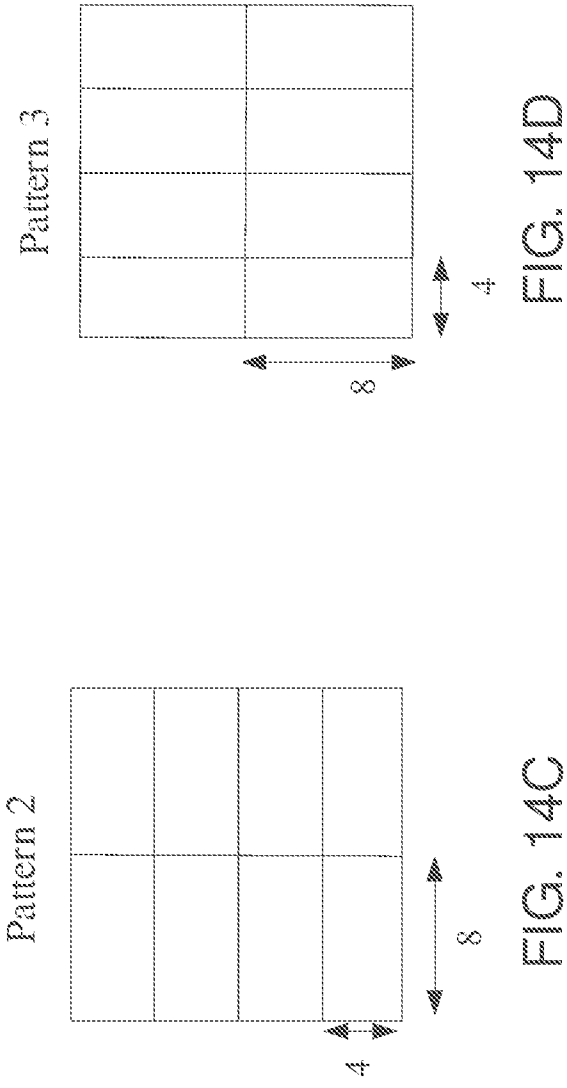
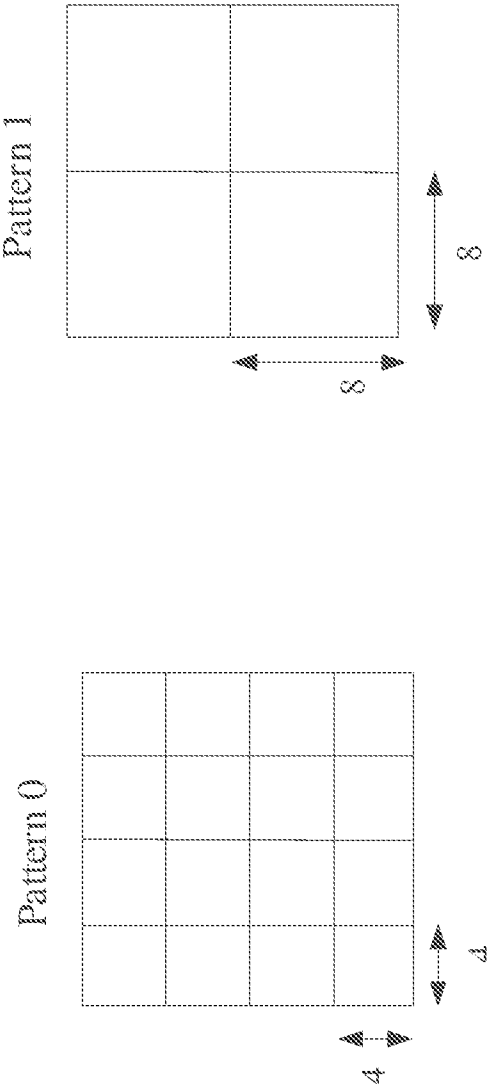


FIG. 13



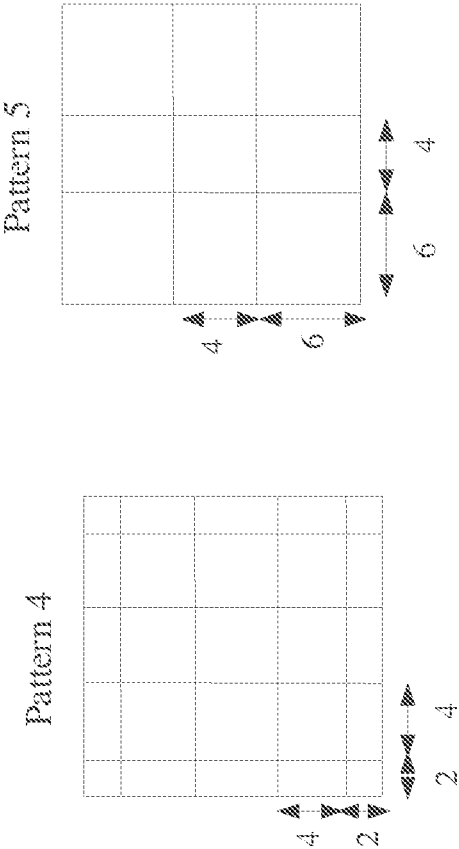


FIG. 14E

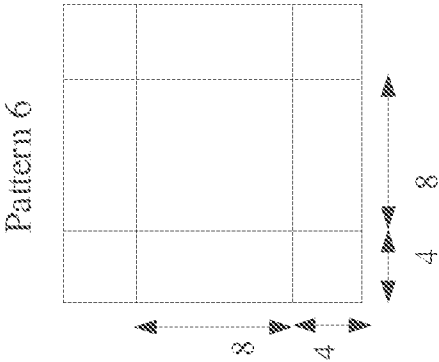
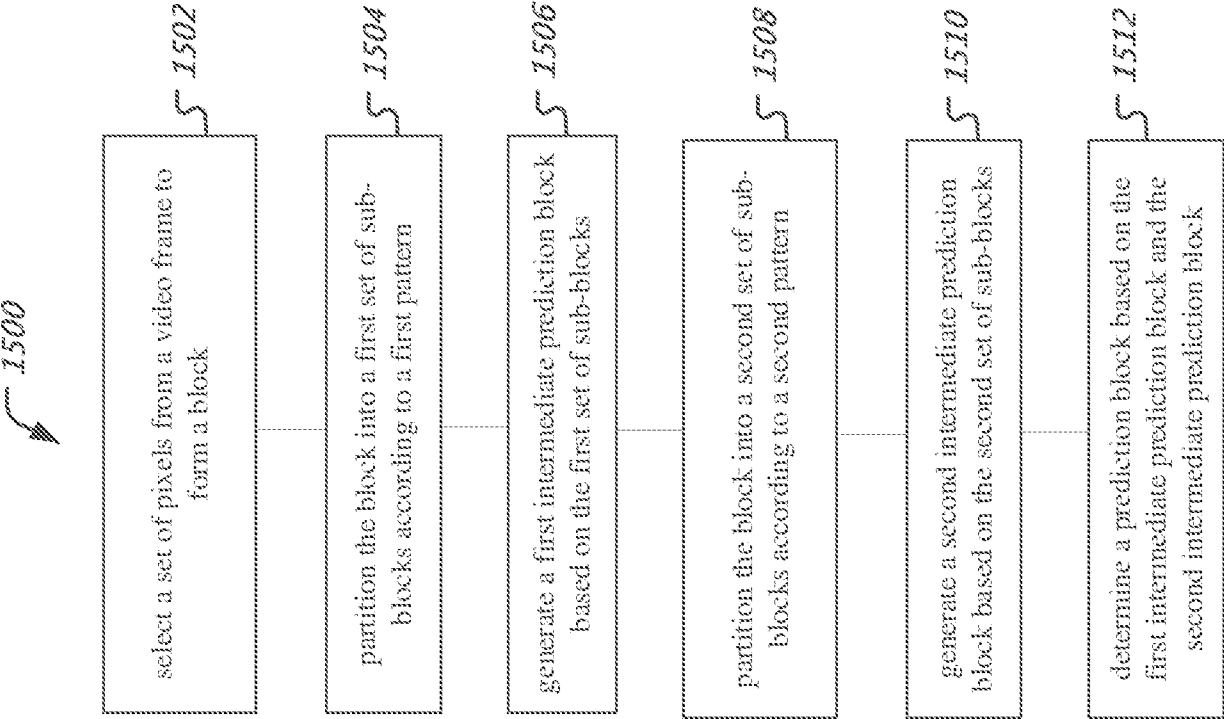


FIG. 14G



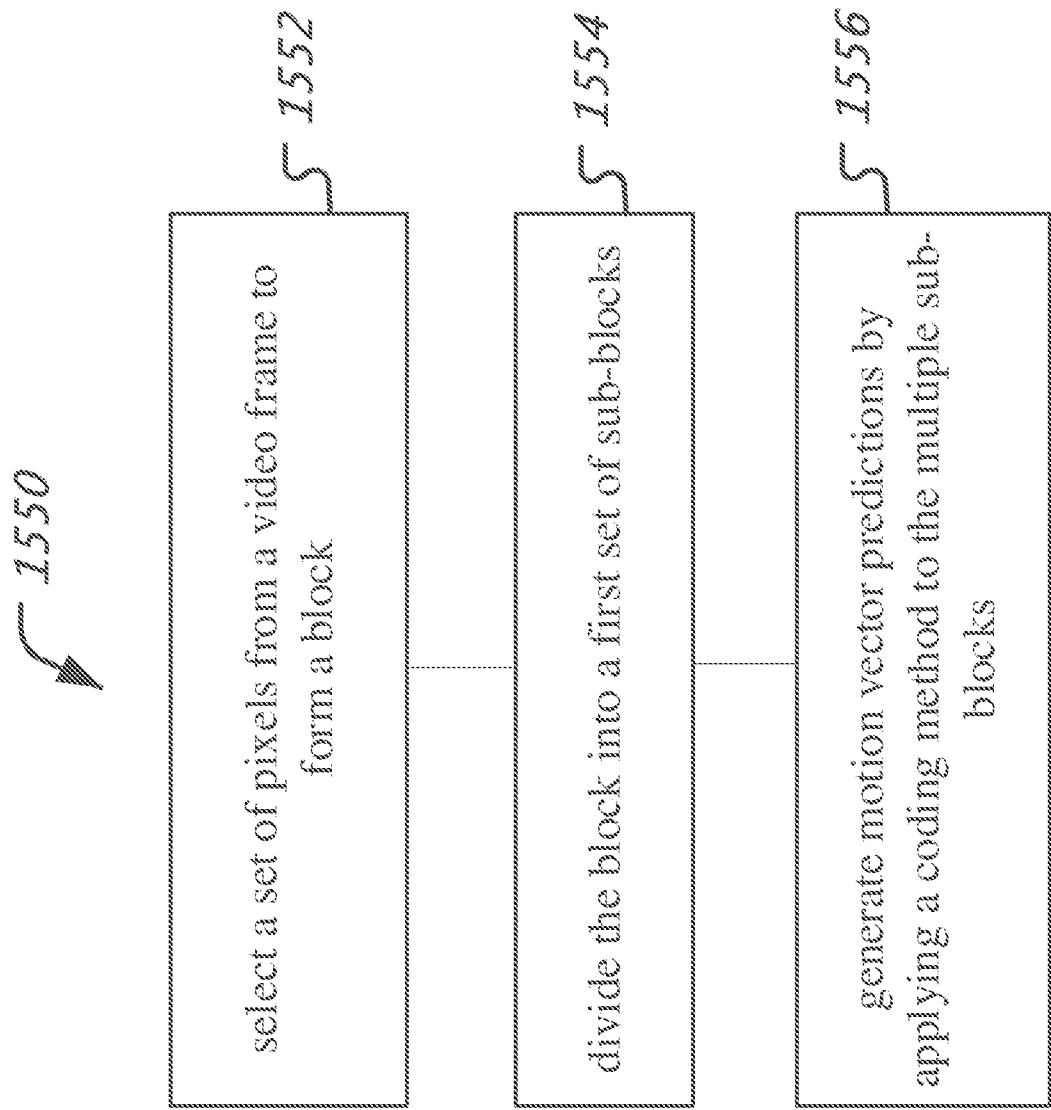


FIG. 15B

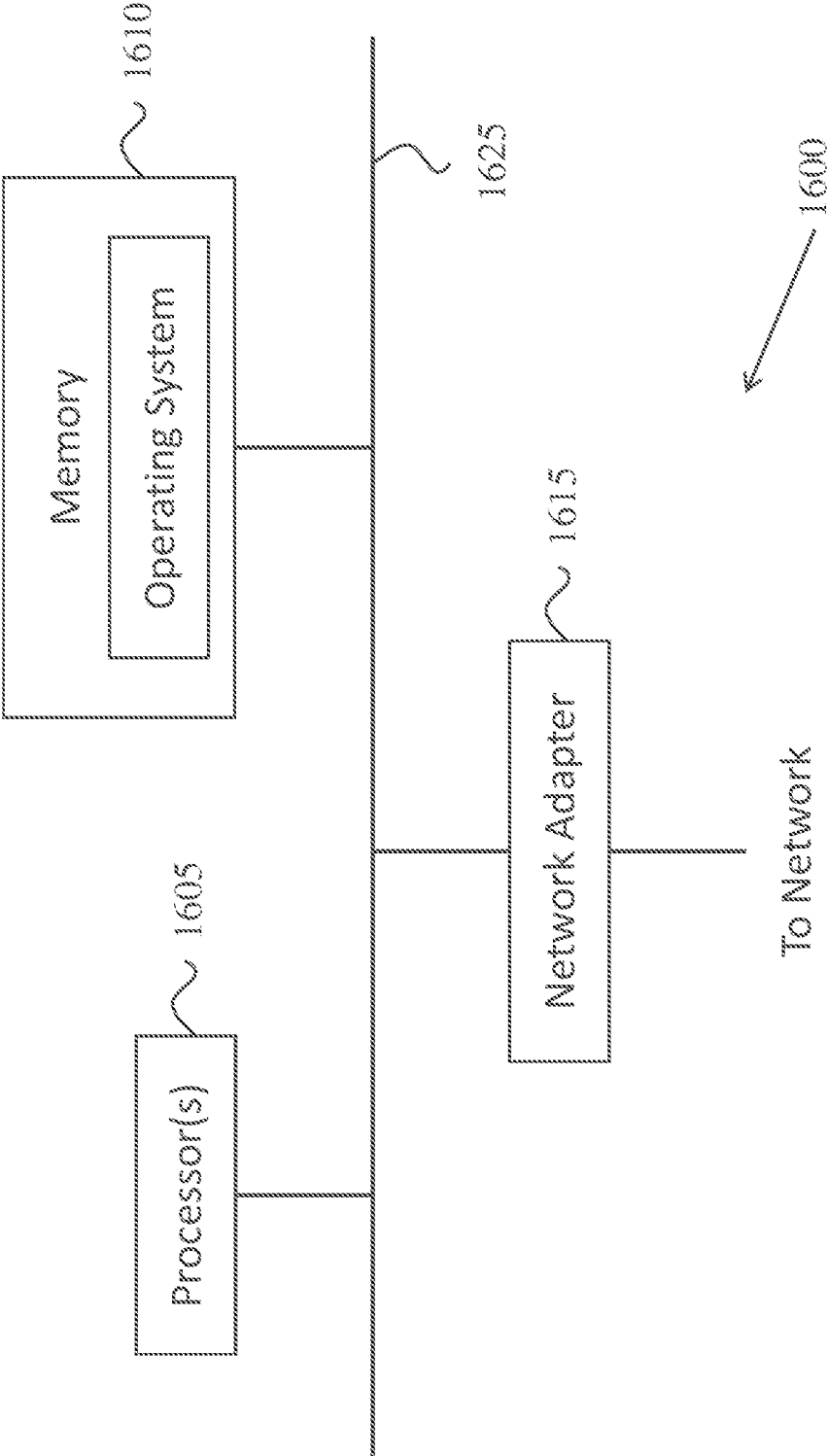


FIG. 16

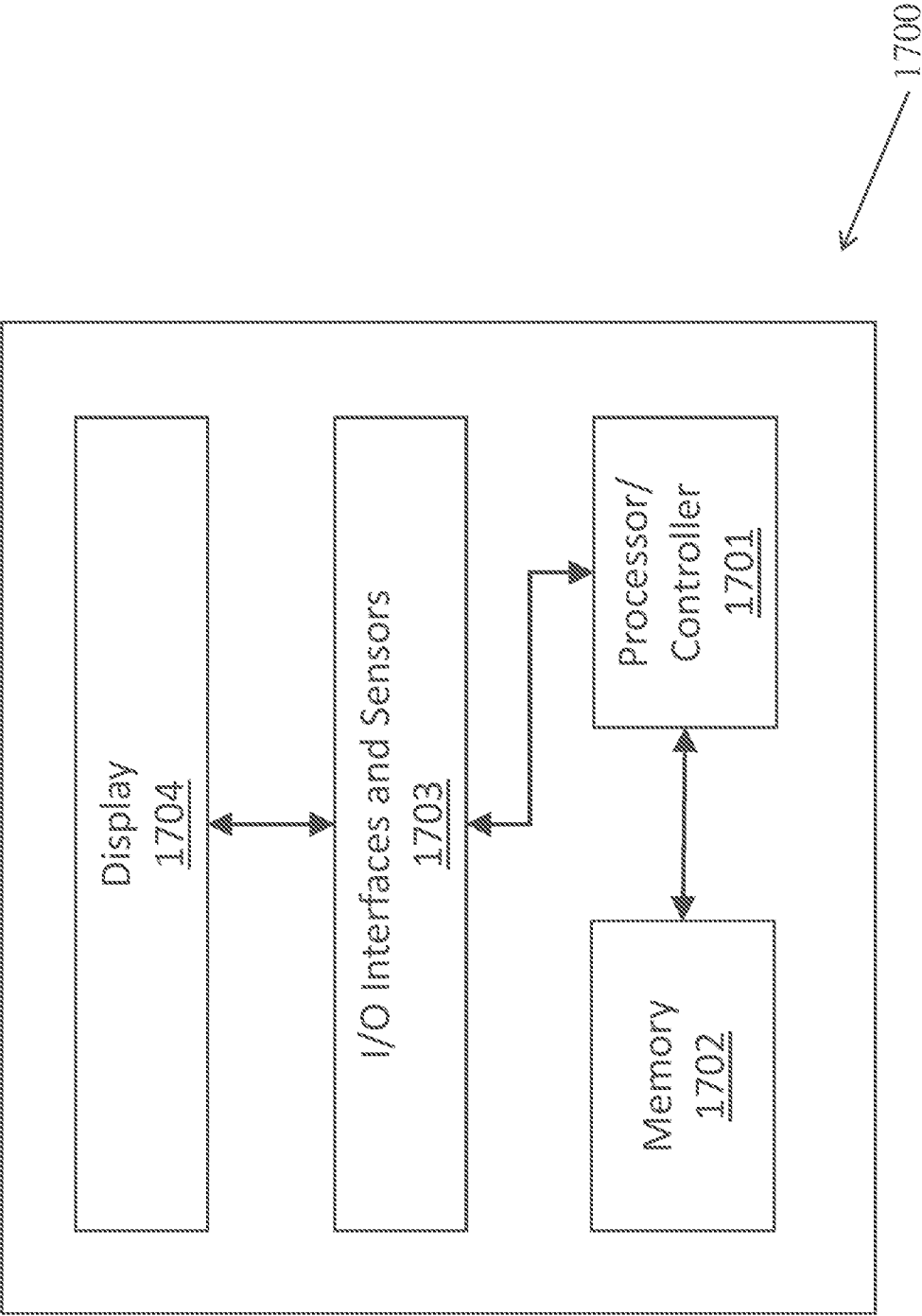


FIG. 17

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2019/054466

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04N19/573 H04N19/577 H04N19/119 H04N19/176 H04N19/105
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	FLIERL M ET AL: "Multihypothesis pictures for H.26L", PROCEEDINGS 2001 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. ICIP 2001 - THESSALONIKI, GREECE, OCT. 7 - 10, 2001; [INTERNATIONAL CONFERENCE ON IMAGE PROCESSING], INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, NEW YORK, NY, vol. 3, 7 October 2001 (2001-10-07), pages 526-529, XP010563399, ISBN: 978-0-7803-6725-8	1-6, 10-30, 33,34
Y A	abstract figures 1, 2 sections 1, 2 and 3 ----- -/-	7-9 32

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

30 August 2019

Date of mailing of the international search report

09/09/2019

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Wedi, Thomas

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2019/054466

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6 807 231 B1 (WIEGAND THOMAS [DE] ET AL) 19 October 2004 (2004-10-19) figure 1 and 2 equation 4	7-9
X	----- CHEN J ET AL: "JVET-G1001- Algorithm description of Joint Exploration Test Model 7 (JEM7)", JOINT VIDEO EXPLORATION TEAM (JVET) OF ITU-T SG 16 WP 3 AND ISO/IEC JTC 1/SC 29/WG 11, 7TH MEETING, 13-7-2017 - 21-7-2017; TORINO , 19 August 2017 (2017-08-19), pages i-iv, 1, XP030150980, Retrieved from the Internet: URL: http://phenix.int-evry.fr/jvet/doc_end_user/documents/7_Torino/wg11/JVET-G0001-v1.zip abstract figure 1 and 2 section 2.1 section 2.3.6 section 2.3.8	1,3-5, 11-14, 16-20, 22,31, 32,34
X,P	----- ZHANG (BYTEDANCE) K ET AL: "CE4-related: Interweaved Prediction for Affine Motion Compensation", 11. JVET MEETING; 20180711 - 20180718; LJUBLJANA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-K0102 3 July 2018 (2018-07-03), XP030198782, Retrieved from the Internet: URL: http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0102-v1.zip JVET-K0102.docx [retrieved on 2018-07-03] sections 1 and 2 -----	1-34

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2019/054466

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6807231	B1	19-10-2004	NONE
