



(51) International Patent Classification:

H04N 19/105 (2014.01) H04N 19/176 (2014.01)

H04N 19/159 (2014.01) H04N 19/537 (2014.01)

(21) International Application Number:

PCT/IB2019/057400

(22) International Filing Date:

03 September 2019 (03.09.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2018/103770

03 September 2018 (03.09.2018) CN

(71) Applicants: BEIJING BYTEDANCE NETWORK

TECHNOLOGY CO., LTD. [CN/CN]; Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District, 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 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, 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,

(54) Title: FAST ENCODING METHODS FOR INTERWEAVED PREDICTION

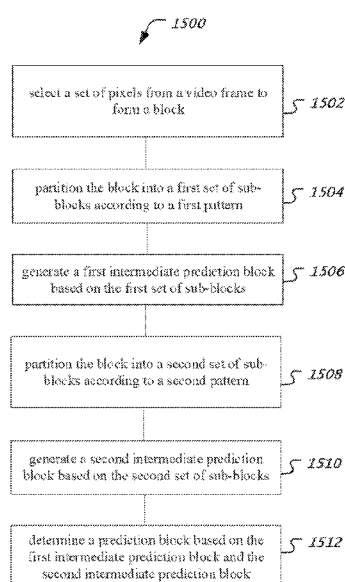


FIG. 15A

(57) Abstract: Methods, systems, and devices related to sub-block based motion prediction in video coding are described. In one representative aspect, a method for video processing includes determining, based on a characteristic of a video block, a prediction block for the video block using a first intermediate prediction block and a second intermediate prediction block, and generating a coded representation of the video block using the prediction block. The first intermediate prediction block is generated by partitioning the video block into a first set of sub-blocks and the second intermediate prediction block is generated by partitioning the video block into a second set of sub-blocks. At least one sub-block in the second set has a different size than a sub-block in the first set..

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*

FAST ENCODING METHODS FOR 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 priority to and benefit of International Patent Application No. PCT/CN2018/103770, filed on September 3, 2018. For all purposes, the entire disclosure of the International Patent Application No. PCT/CN2018/103770 is incorporated by reference as part of the disclosure of this patent document.

TECHNICAL FIELD

[0002] This patent document relates to video processing 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 for video processing is disclosed. The method includes determining, based on a characteristic of a video block, a prediction block for the video block using a first intermediate prediction block and a second intermediate prediction block, and generating a coded representation of the video block using the prediction block. The first intermediate prediction block is generated by partitioning the video block into a first set of sub-blocks and the second intermediate prediction block is generated by partitioning the video block into a second set of sub-blocks. 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, a method for improving bandwidth usage and

prediction accuracy of a block-based motion prediction video system is disclosed. The method includes selecting a set of pixels from a video frame to form a block, partitioning the block into a first set of sub-blocks according to a first pattern, generating a first intermediate prediction block based on the first set of sub-blocks, partitioning the block into a second set of sub-blocks according to a second pattern, generating a second intermediate prediction block based on the second set of sub-blocks, and determining a prediction block based on the first intermediate prediction block and the second intermediate prediction block. At least one sub-block in the second set has a different size than a sub-block in the first set.

[0007] In another representative aspect, a method for improving block-based motion prediction in a video system is disclosed. The method includes selecting a set of pixels from a video frame to form a block, 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, and generating motion vector predictions by applying a coding algorithm to the multiple sub-blocks. At least one sub-block of the multiple sub-blocks has a different size than other sub-blocks

[0008] In another representative aspect, an apparatus comprising a processor and a non-transitory memory with instructions thereon is disclosed. The instructions, upon execution by the processor, cause the processor to select a set of pixels from a video frame to form a block, partition the block into a first set of sub-blocks according to a first pattern, generate a first intermediate prediction block based on the first set of sub-blocks, partition the block into a second set of sub-blocks according to a second pattern, wherein at least one sub-block in the second set has a different size than a sub-block in the first set, generate a second intermediate prediction block based on the second set of sub-blocks, and determine a prediction block based on the first intermediate prediction block and the second intermediate prediction block.

[0009] 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.

[0010] In yet another representative aspect, a video decoder apparatus may implement a method as described herein.

[0011] The details of one or more implementations are set forth in the accompanying attachments, the drawings, and the description below. Other features will be apparent from the

description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0012] FIG. 1 is a schematic diagram showing an example of sub-block based prediction.
- 5 [0013] FIG. 2 shows an example of an affine motion field of a block described by two control point motion vectors.
- [0014] FIG. 3 shows an example of affine motion vector field per sub-block for a block.
- [0015] FIG. 4 shows an example of motion vector prediction for a block in the AF_INTER mode.
- 10 [0016] FIG. 5A shows an example of the selection order of candidate blocks for a current Coding Unit (CU).
- [0017] FIG. 5B shows another example of candidate blocks for a current CU in the AF_MERGE mode.
- [0018] FIG. 6 shows an example of Alternative Temporal Motion Vector Prediction (ATMVP) motion prediction process for a CU.
- 15 [0019] FIG. 7 shows an example of one CU with four sub-blocks and neighboring blocks.
- [0020] FIG. 8 shows an example optical flow trajectory in the Bi-directional Optical flow (BIO) method.
- [0021] FIG. 9A shows an example of access positions outside of a block.
- 20 [0022] FIG. 9B shows that a padding area can be used to avoid extra memory access and calculation.
- [0023] FIG. 10 shows an example of bilateral matching used in the Frame-Rate Up Conversion (FRUC) method.
- [0024] FIG. 11 shows an example of template matching used in the FRUC method.
- 25 [0025] FIG. 12 shows an example of unilateral Motion Estimation (ME) in the FRUC method.
- [0026] FIG. 13 shows an example of interweaved prediction with two dividing patterns in accordance with the disclosed technology.
- [0027] FIG. 14A shows an example dividing pattern in which block is divided into 4×4 sub-
- 30 blocks in accordance with the disclosed technology.
- [0028] FIG. 14B shows an example dividing pattern in which a block is divided into 8×8

sub-blocks in accordance with the disclosed technology.

[0029] FIG. 14C shows an example dividing pattern in which a block is divided into 4×8 sub-blocks in accordance with the disclosed technology.

[0030] FIG. 14D shows an example dividing pattern in which a block is divided into 8×4 sub-blocks in accordance with the disclosed technology.

[0031] FIG. 14E shows an example dividing pattern in which a block is divided into non-uniform sub-blocks in accordance with the disclosed technology.

[0032] FIG. 14F shows another example dividing pattern in which a block is divided into non-uniform sub-blocks in accordance with the disclosed technology.

[0033] 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.

[0034] 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.

[0035] 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.

[0036] 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.

[0037] 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.

[0038] FIG. 18A-18C show example embodiments of a partial interweaved prediction technique.

[0039] FIG. 19 is a flowchart representation of a method for improving block-based motion prediction in a video system.

[0040] FIG. 20 is a block diagram of an example video processing system in which disclosed techniques may be implemented.

DETAILED DESCRIPTION

[0041] Global motion compensation is one of variations of motion compensation techniques

and can be used for predicting camera's motion. However, moving objects within a frame 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.

[0042] 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). 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.

[0043] 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.

[0044] Affine Prediction

[0045] 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. 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:

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

[0047] As shown in FIG. 2, (v_{0x}, v_{0y}) is motion vector of the top-left corner control point, and

(v_{1x}, v_{1y}) is 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:

$$[0048] \quad \begin{cases} M = \text{clip3}(4, w, \frac{w \times MvPre}{\max(\text{abs}(v_{1x} - v_{0x}), \text{abs}(v_{1y} - v_{0y}))}) \\ N = \text{clip3}(4, h, \frac{h \times MvPre}{\max(\text{abs}(v_{2x} - v_{0x}), \text{abs}(v_{2y} - v_{0y}))}) \end{cases} \quad \text{Eq. (2)}$$

5 [0049] Here, $MvPre$ is the motion vector fraction accuracy (e.g., 1/16 in JEM). (v_{2x}, v_{2y}) is 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.

[0050] 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.

15 [0051] 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. 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. If 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.

[0052] 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.

[0053] 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.

[0054] Alternative Temporal Motion Vector Prediction (ATMVP)

[0055] 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.

[0056] 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.

[0057] 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.

[0058] 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.

[0059] Spatial Temporal Motion Vector Prediction (STMVP)

[0060] 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).

[0061] 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.

[0062] Bi-directional Optical flow (BIO)

[0063] The Bi-directional Optical flow (BIO) method is sample-wise motion refinement performed on top of block-wise motion compensation for bi-prediction. In some implementations, the sample-level motion refinement does not use signaling.

[0064] Let $I^{(k)}$ be the luma value from reference k ($k=0, 1$) after block motion compensation, and $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ are horizontal and vertical components of the $I^{(k)}$ gradient, respectively. Assuming the optical flow is valid, the motion vector field (v_x, v_y) is given by:

$$\partial I^{(k)}/\partial t + v_x \partial I^{(k)}/\partial x + v_y \partial I^{(k)}/\partial y = 0. \quad \text{Eq. (3)}$$

[0066] Combining this optical flow equation with Hermite interpolation for the motion trajectory of each sample results in a unique third-order polynomial that matches both the function values $I^{(k)}$ and derivatives $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ at the ends. The value of this polynomial at $t=0$ is the BIO prediction:

$$pred_{BIO} = 1/2 \cdot (I^{(0)} + I^{(1)} + v_x/2 \cdot (\tau_1 \partial I^{(1)}/\partial x - \tau_0 \partial I^{(0)}/\partial x) + v_y/2 \cdot (\tau_1 \partial I^{(1)}/\partial y - \tau_0 \partial I^{(0)}/\partial y)) \quad \text{Eq. (4)}$$

[0068] FIG. 8 shows an example optical flow trajectory in the Bi-directional Optical flow (BIO) method. Here, τ_0 and τ_1 denote the distances to the reference frames. Distances τ_0 and τ_1 are calculated based on POC for Ref₀ and Ref₁: $\tau_0 = \text{POC}(\text{current}) - \text{POC}(\text{Ref}_0)$, $\tau_1 = \text{POC}(\text{Ref}_1) - \text{POC}(\text{current})$. If both predictions come from the same time direction (either both from the past or both from the future) then the signs are different (e.g., $\tau_0 \cdot \tau_1 < 0$). In this case, BIO is applied if the prediction is not from the same time moment (e.g., $\tau_0 \neq \tau_1$). Both referenced regions have non-zero motion (e.g., $MV_{x_0}, MV_{y_0}, MV_{x_1}, MV_{y_1} \neq 0$) and the block motion vectors are proportional to the time distance (e.g., $MV_{x_0}/MV_{x_1} = MV_{y_0}/MV_{y_1} = -\tau_0/\tau_1$).

[0069] The motion vector field (v_x, v_y) is determined by minimizing the difference

Δ between values in points A and B. FIGS. 9A-9B show an example of intersection of motion trajectory and reference frame planes. Model uses only first linear term of a local Taylor expansion for Δ :

$$[0070] \quad \Delta = (I^{(0)} - I^{(1)})_0 + v_x (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) + v_y (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \quad \text{Eq. (5)}$$

- 5 [0071] All values in the above equation depend on the sample location, denoted as (i', j') . Assuming the motion is consistent in the local surrounding area, Δ can be minimized inside the $(2M+1) \times (2M+1)$ square window Ω centered on the currently predicted point (i, j) , where M is equal to 2:

$$[0072] \quad (v_x, v_y) = \underset{v_x, v_y}{\operatorname{argmin}} \sum_{[i', j'] \in \Omega} \Delta^2[i', j'] \quad \text{Eq. (6)}$$

- 10 [0073] For this optimization problem, the JEM uses a simplified approach making first a minimization in the vertical direction and then in the horizontal direction. This results in the following:

$$[0074] \quad v_x = (s_1 + r) > m? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_3}{(s_1 + r)} \right) : 0 \quad \text{Eq. (7)}$$

$$[0075] \quad v_y = (s_5 + r) > m? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_6 - v_x s_2 / 2}{(s_5 + r)} \right) : 0 \quad \text{Eq. (8)}$$

- 15 [0076] where,

$$[0077] \quad \begin{aligned} s_1 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_3 = \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x), \\ s_2 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y), \\ s_5 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_6 = \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad \text{Eq. (9)}$$

[0078] In order to avoid division by zero or a very small value, regularization parameters r and m can be introduced in Eq. (7) and Eq. (8).

$$[0079] \quad r = 500 \cdot 4^{d-8} \quad \text{Eq. (10)}$$

$$20 \quad [0080] \quad m = 700 \cdot 4^{d-8} \quad \text{Eq. (11)}$$

[0081] Here, d is bit depth of the video samples.

[0082] In order to keep the memory access for BIO the same as for regular bi-predictive motion compensation, all prediction and gradients values, $I^{(k)}$, $\partial I^{(k)} / \partial x$, $\partial I^{(k)} / \partial y$, are calculated for positions inside the current block. FIG. 9A shows an example of access positions outside of a block 900. As shown in FIG. 9A, in Eq. (9), $(2M+1) \times (2M+1)$ square window Ω

25

centered in currently predicted point on a boundary of predicted block needs to accesses positions outside of the block. In the JEM, values of $I^{(k)}$, $\partial I^{(k)} / \partial x$, $\partial I^{(k)} / \partial y$ outside of the block are set to be equal to the nearest available value inside the block. For example, this can be implemented as a padding area 901, as shown in FIG. 9B.

- 5 **[0083]** With BIO, it is possible that the motion field can be refined for each sample. To reduce the computational complexity, a block-based design of BIO is used in the JEM. The motion refinement can be calculated based on a 4x4 block. In the block-based BIO, the values of s_n in Eq. (9) of all samples in a 4x4 block can be aggregated, and then the aggregated values of s_n in are used to derived BIO motion vectors offset for the 4x4 block. More specifically, the
- 10 following formula can used for block-based BIO derivation:

$$\begin{aligned}
 s_{1,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega(x,y)} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_{3,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x), \\
 s_{2,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y), \\
 s_{5,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_{6,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)
 \end{aligned}
 \tag{12}$$

[0085] Here, b_k denotes the set of samples belonging to the k-th 4x4 block of the predicted block. s_n in Eq (7) and Eq (8) are replaced by $((s_{n,b_k}) \gg 4)$ to derive the associated motion vector offsets.

- 15 **[0086]** In some scenarios, MV regiment of BIO may be unreliable due to noise or irregular motion. Therefore, in BIO, the magnitude of MV regiment is clipped to a threshold value. The threshold value is determined based on whether the reference pictures of the current picture are all from one direction. For example, if all the reference pictures of the current picture are from one direction, the value of the threshold is set to $12 \times 2^{14-d}$; otherwise, it is set to $12 \times 2^{13-d}$.

- 20 **[0087]** Gradients for BIO can be calculated at the same time with motion compensation interpolation using operations consistent with HEVC motion compensation process (e.g., 2D separable Finite Impulse Response (FIR)). In some embodiments, the input for the 2D separable FIR is the same reference frame sample as for motion compensation process and fractional position $(fracX, fracY)$ according to the fractional part of block motion vector. For horizontal
- 25 gradient $\partial I / \partial x$, a signal is first interpolated vertically using *BIOfilterS* corresponding to the fractional position $fracY$ with de-scaling shift $d-8$. Gradient filter *BIOfilterG* is then applied in horizontal direction corresponding to the fractional position $fracX$ with de-scaling shift by $18-d$. For vertical gradient $\partial I / \partial y$, a gradient filter is applied vertically using *BIOfilterG* corresponding

to the fractional position $fracY$ with de-scaling shift $d-8$. The signal displacement is then performed using $BIOfilterS$ in horizontal direction corresponding to the fractional position $fracX$ with de-scaling shift by $18-d$. The length of interpolation filter for gradients calculation $BIOfilterG$ and signal displacement $BIOfilterF$ can be shorter (e.g., 6-tap) in order to maintain reasonable complexity. Table 1 shows example filters that can be used for gradients calculation of different fractional positions of block motion vector in BIO. Table 2 shows example interpolation filters that can be used for prediction signal generation in BIO.

Table 1 Example filters for gradient calculation in BIO

Fractional pel position	Interpolation filter for gradient($BIOfilterG$)
0	{ 8, -39, -3, 46, -17, 5 }
1/16	{ 8, -32, -13, 50, -18, 5 }
1/8	{ 7, -27, -20, 54, -19, 5 }
3/16	{ 6, -21, -29, 57, -18, 5 }
1/4	{ 4, -17, -36, 60, -15, 4 }
5/16	{ 3, -9, -44, 61, -15, 4 }
3/8	{ 1, -4, -48, 61, -13, 3 }
7/16	{ 0, 1, -54, 60, -9, 2 }
1/2	{ -1, 4, -57, 57, -4, 1 }

Table 2 Example interpolation filters for prediction signal generation in BIO

Fractional pel position	Interpolation filter for prediction signal(BIOfilterS)
0	{ 0, 0, 64, 0, 0, 0 }
1/16	{ 1, -3, 64, 4, -2, 0 }
1/8	{ 1, -6, 62, 9, -3, 1 }
3/16	{ 2, -8, 60, 14, -5, 1 }
1/4	{ 2, -9, 57, 19, -7, 2 }
5/16	{ 3, -10, 53, 24, -8, 2 }
3/8	{ 3, -11, 50, 29, -9, 2 }
7/16	{ 3, -11, 44, 35, -10, 3 }
1/2	{ 3, -10, 35, 44, -11, 3 }

[0088] In the JEM, BIO can be applied to all bi-predicted blocks when the two predictions are from different reference pictures. When Local Illumination Compensation (LIC) is enabled for a CU, BIO can be disabled.

[0089] In some embodiments, OBMC is applied for a block after normal MC process. To reduce the computational complexity, BIO may not be applied during the OBMC process. This means that BIO is applied in the MC process for a block when using its own MV and is not applied in the MC process when the MV of a neighboring block is used during the OBMC process.

[0090] Frame-Rate Up Conversion (FRUC)

[0091] 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.

[0092] 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.

5 [0093] 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
10 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.

[0094] For example, the following derivation process is performed for a $W \times H$ CU motion
15 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.

$$[0095] \quad M = \max\{4, \min\{\frac{M}{2^D}, \frac{N}{2^D}\}\} \quad \text{Eq. (13)}$$

20 [0096] 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
25 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.

30 [0097] 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.

[0098] 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.

[0099] 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 .

[00100] 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.

[00101] 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.

[00102] *Generation of interpolated MV field*

[00103] 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.

[00104] 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.

[00105] *Interpolation and matching cost*

[00106] 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.

[00107] 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 determined, the matching cost C of bilateral matching at sub-CU level search is calculated as follows:

$$\text{[00108]} \quad C = SAD + w \cdot (|MV_x - MV_x^s| + |MV_y - MV_y^s|) \quad \text{Eq. (14)}$$

[00109] Here, w is a weighting factor. In some embodiments, w can be empirically set to 4. 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.

[00110] 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.

[00111] 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 luma sample MV refinement. The search range of MV refinement for the CU and sub-CU step are set equal to 8 luma samples.

[00112] 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 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:

[00113] If $\text{costBi} \leq \text{factor} * \min(\text{cost0}, \text{cost1})$

[00114] bi-prediction is used;

[00115] Otherwise, if $\text{cost0} \leq \text{cost1}$

[00116] uni-prediction from list0 is used;

[00117] Otherwise,

[00118] uni-prediction from list1 is used;

[00119] 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. The inter prediction direction selection can be applied to the CU-level template matching process.

[00120] 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. 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.

[00121] 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.

[00122] 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.

[00123] 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).

[00124] 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

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

[00126] 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:

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

[00128] N is a non-negative value. Alternatively, the bit-shifting operation in Eq. (16) can also be expressed as:

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

[00130] 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.

[00131] 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.

[00132] 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. 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).

[00133] 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 .

[00134] In some embodiments, whether to apply interweaved prediction can be determined based on the inter-prediction direction. For example, in some embodiments, the interweaved prediction may be applied for bi-prediction but not for uni-prediction. 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.

[00135] 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. 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

$$[00136] \quad P(x, y) = \frac{w^0(x, y) \times P^0(x, y) + w^1(x, y) \times P^1(x, y)}{w^0(x, y) + w^1(x, y)} \quad \text{Eq. (18)}$$

[00137] 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: $w^0(x, y) + w^1(x, y) = 1 \ll N$ (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.

[00138] 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

$$[00139] \quad P^L(x, y) = \frac{\sum_{i=0}^{XL-1} w_i^L(x, y) \times P_i^L(x, y)}{\sum_{i=0}^{XL-1} w_i^L(x, y)} \quad \text{Eq. (19)}$$

[00140] Here XL is the number of dividing patterns for list L . $P_i^L(x, y)$ is the prediction 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. 14D.

[00141] In some embodiments, a bi-predicted block with sub-block based prediction is 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

$$[00142] \quad P(x, y) = \frac{a * \frac{\sum_{i=0}^{XL0-1} w_i^0(x, y) \times P_i^0(x, y)}{\sum_{i=0}^{XL0-1} w_i^0(x, y)} + b * \frac{\sum_{i=0}^{XL1-1} w_i^1(x, y) \times P_i^1(x, y)}{\sum_{i=0}^{XL1-1} w_i^1(x, y)}}{a+b} \quad \text{Eq. (20)}$$

[00143] Here parameters a and b are two additional weights applied to the two internal 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.

[00144] 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.

[00145] 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.

[00146] 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, 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), a CTU/LCU, a CU, a PU, a TU, or a first block of a region.

[00147] 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.

[00148] 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).

[00149] 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).

[00150] 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.

[00151] 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.

[00152] 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.

[00153] 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.

[00154] 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.

[00155] 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.

[00156] 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, partitioning the block into the second set of sub-blocks is performed for a motion prediction of the block in a second direction.

[00157] 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 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 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.

[00158] 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 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.

[00159] In some embodiments, determining the prediction result includes applying a first set 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.

[00160] In some embodiments, the first set of weights or the second set of weights includes 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.

[00161] 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.

[00162] 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

5 (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.

[00163] In the methods 1500 and 1550, partial interweaving may be implemented. Using this scheme, samples in a first subset of prediction samples are calculated as a weighted combination

10 of the first intermediate prediction block and samples a second subset of the prediction samples are copied from sub-blocked based prediction wherein the first subset and the second subset are based on a dividing pattern. The first subset and the second subset may together make up the entire prediction block, e.g., the block that is currently being processed. As depicted in FIGs. 18A to 18C, in various examples, the second subset that is excluded from interweaving could be

15 made up of (a) corner sub-blocks or (b) uppermost and lowermost row of sub-blocks or (c) left-most or right-most columns of sub-blocks. The size of the block being currently processed may be used as a condition for deciding whether to exclude certain sub-blocks from interweaved prediction. For example, certain conditions are described in paragraphs below.

[00164] As further described in the present document, the encoding process may refrain from

20 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.

[00165] 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

25 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

30 Electronics Engineers (IEEE) standard 674 bus, sometimes referred to as "Firewire."

[00166] 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
5 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.

[00167] 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
10 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.

[00168] Also connected to the processor(s) 1605 through the interconnect 1625 is a (optional)
15 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.

[00169] 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
20 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
25 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
30 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.

[00170] 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 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.

[00171] 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 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.

[00172] 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. 16 and FIG. 17.

[00173] Partial interweaving

[00174] In some embodiments, partial interweaved prediction may be achieved as follow.

[00175] In some embodiments, interweaved prediction is applied to a part of the current block. Prediction samples at some positions are calculated as the weighted sum of two or more sub-block based predictions. Prediction samples at other positions are not used for the weighted sum. For example, these prediction samples are copied from the sub-block based prediction with a certain dividing pattern.

[00176] In some embodiments, the current block is predicted by sub-block based prediction P1 and P2 with dividing pattern D0 and dividing pattern D1, respectively. The final prediction is calculated as $P = w_0 \times P_0 + w_1 \times P_1$. At some positions, $w_0 \neq 0$ and $w_1 \neq 0$. But at some other positions, $w_0 = 1$ and $w_1 = 0$, that is, interweaved prediction is not applied at those positions.

[00177] In some embodiments, interweaved prediction is not applied on four corner sub-blocks as shown in Fig. 18A.

[00178] In some embodiments, interweaved prediction is not applied on the left-most column of sub-blocks and right-most column of sub-blocks as shown in FIG. 18B.

[00179] In some embodiments, interweaved prediction is not applied on the top-most row of sub-blocks and bottom-most row of sub-blocks as shown in Fig. 18C.

[00180] Examples of Techniques incorporated within Encoder Embodiments

[00181] In some embodiments, interweaved prediction is not applied in the motion estimation (ME) process.

[00182] For example, interweaved prediction is not applied in the ME process for the 6-parameter affine prediction.

[00183] For example, interweaved prediction is not applied in the ME process if the size of

the current block satisfies certain conditions such as follows. Here, it is assumed that the width and height of the current block is W and H respectively and T , T_1 , T_2 are integer values:

[00184] $W \geq T_1$ and $H \geq T_2$;

[00185] $W \leq T_1$ and $H \leq T_2$;

5 [00186] $W \geq T_1$ or $H \geq T_2$;

[00187] $W \leq T_1$ or $H \leq T_2$;

[00188] $W+H \geq T$

[00189] $W+H \leq T$

[00190] $W \times H \geq T$

10 [00191] $W \times H \leq T$

[00192] For example, interweaved prediction is omitted in the ME process if the current block is split from a parent block, and the parent block does not choose affine mode at encoder.

[00193] FIG. 19 is a flowchart representation of a method 1900 for improving block-based motion prediction in a video system. The method 1900 includes, at operation 1902, determining, based on a characteristic of a video block, a prediction block for the video block using a first intermediate prediction block and a second intermediate prediction block. The method 1900 includes, at operation 1904, generating a coded representation of the video block using the prediction block. The first intermediate prediction block is generated by partitioning the video block into a first set of sub-blocks and the second intermediate prediction block is generated by partitioning the video block into a second set of sub-blocks. At least one sub-block in the second set has a different size than a sub-block in the first set.

[00194] In some embodiments, the determining is performed at a stage of motion estimation in an encoding process. In some embodiments, the characteristic of the video block indicates that the prediction block is not determined based on an affine prediction.

25 [00195] In some embodiments, the block has a width of W and a height of H , and the characteristic of the video block indicates that the prediction block is determined due to W and H not satisfying one or more conditions. In some embodiments, the one or more conditions include $W \geq T_1$ and $H \geq T_2$, T_1 and T_2 being predefined integer values. In some embodiments, the one or more conditions include $W \leq T_1$ and $H \leq T_2$, T_1 and T_2 being predefined integer values. In some embodiments, the one or more conditions include $W \geq T_1$ or $H \geq T_2$, T_1 and T_2 being predefined integer values. In some embodiments, the one or more conditions include $W \leq T_1$ or

$H \leq T_2$, T_1 and T_2 being predefined integer values. In some embodiments, the one or more conditions include $W+H \geq T$, T being a predefined integer value. In some embodiments, the one or more conditions include $W+H \leq T$, T being a predefined integer value. In some embodiments, the one or more conditions include $W \times H \geq T$, T being a predefined integer value. In some

5 embodiments, the one or more conditions include $W \times H \leq T$, T being a predefined integer value.

[00196] In some embodiments, the characteristic of the video block indicates that the video block is not split from a parent block. In some embodiments, the characteristic of the video block indicates that the block is split from a parent block that is affine coded.

[00197] Alternatively, affine mode is not checked at encoder if the current block is split from

10 a parent block, and the parent block does not choose affine mode at encoder.

[00198] FIG. 20 is a block diagram showing an example video processing system 2000 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 2000. The system 2000 may include input 2002 for receiving video content. The video content may be received in a raw or uncompressed

15 format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 2002 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

[00199] The system 2000 may include a coding component 2004 that may implement the

20 various coding or encoding methods described in the present document. The coding component 2004 may reduce the average bitrate of video from the input 2002 to the output of the coding component 2004 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 2004 may be either stored, or transmitted via a communication connected, as

25 represented by the component 2006. The stored or communicated bitstream (or coded) representation of the video received at the input 2002 may be used by the component 2008 for generating pixel values or displayable video that is sent to a display interface 2010. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as

30 “coding” operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding

will be performed by a decoder.

[00200] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on.

Examples of storage interfaces include SATA (serial advanced technology attachment), PCI,

5 IDE interface, and the like. The techniques described in the present document may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[00201] 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
10 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.

[00202] 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
15 structural equivalents, or in combinations of one or more of them. The disclosed and other 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
20 memory device, a composition of matter effecting a machine-readable propagated signal, or a 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
25 program in question, e.g., code that constitutes processor firmware, a protocol stack, a database 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.

30 **[00203]** 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

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 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.

[00204] The processes and logic flows described in this document can be performed by one or 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).

[00205] 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.

[00206] 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.

[00207] 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.

[00208] 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 patent document.

CLAIMS

What is claimed is:

1. A method for processing video, comprising:
5 determining, based on a characteristic of a video block, a prediction block for the video block using a first intermediate prediction block and a second intermediate prediction block; and generating a coded representation of the video block using the prediction block, wherein the first intermediate prediction block is generated by partitioning the video block into a first set of sub-blocks, wherein the second intermediate prediction block is generated
10 by partitioning the video block into a second set of sub-blocks, and wherein at least one sub-block in the second set has a different size than a sub-block in the first set.
2. The method of claim 1, wherein the determining is performed at a stage of motion estimation in an encoding process.
15
3. The method of claim 1, wherein the characteristic of the video block indicates that the prediction block is not determined based on an affine prediction.
4. The method of claim 1 or 2, wherein the video block has a width of W and a height of H ,
20 and wherein the characteristic of the video block indicates that the prediction block is determined due to W and H not satisfying one or more conditions.
5. The method of claim 4, wherein the one or more conditions include $W \geq T_1$ and $H \geq T_2$, T_1 and T_2 being predefined integer values.
25
6. The method of claim 4, wherein the one or more conditions include $W \leq T_1$ and $H \leq T_2$, T_1 and T_2 being predefined integer values.
7. The method of claim 4, wherein the one or more conditions include $W \geq T_1$ or $H \geq T_2$, T_1
30 and T_2 being predefined integer values.

8. The method of claim 4, wherein the one or more conditions include $W \leq T_1$ or $H \leq T_2$, T_1 and T_2 being predefined integer values.

9. The method of claim 4, wherein the one or more conditions include $W+H \geq T$, T being a
5 predefined integer value.

10. The method of claim 4, wherein the one or more conditions include $W+H \leq T$, T being a predefined integer value.

10 11. The method of claim 4, wherein the one or more conditions include $W \times H \geq T$, T being a predefined integer value.

12. The method of claim 4, wherein the one or more conditions include $W \times H \leq T$, T being a predefined integer value.

15 13. The method of claim 1 or 2, wherein the characteristic of the video block indicates that the video block is not split from a parent block.

14. The method of claim 1 or 2, wherein the characteristic of the video block indicates that
20 the video block is split from a parent block that is affine coded.

15. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method recited in one or more of claims 1 to 14.

25 16. A non-transitory computer readable media comprising computer program code stored thereon, the computer program code is for carrying out the method recited in one or more of claims 1 to 14.

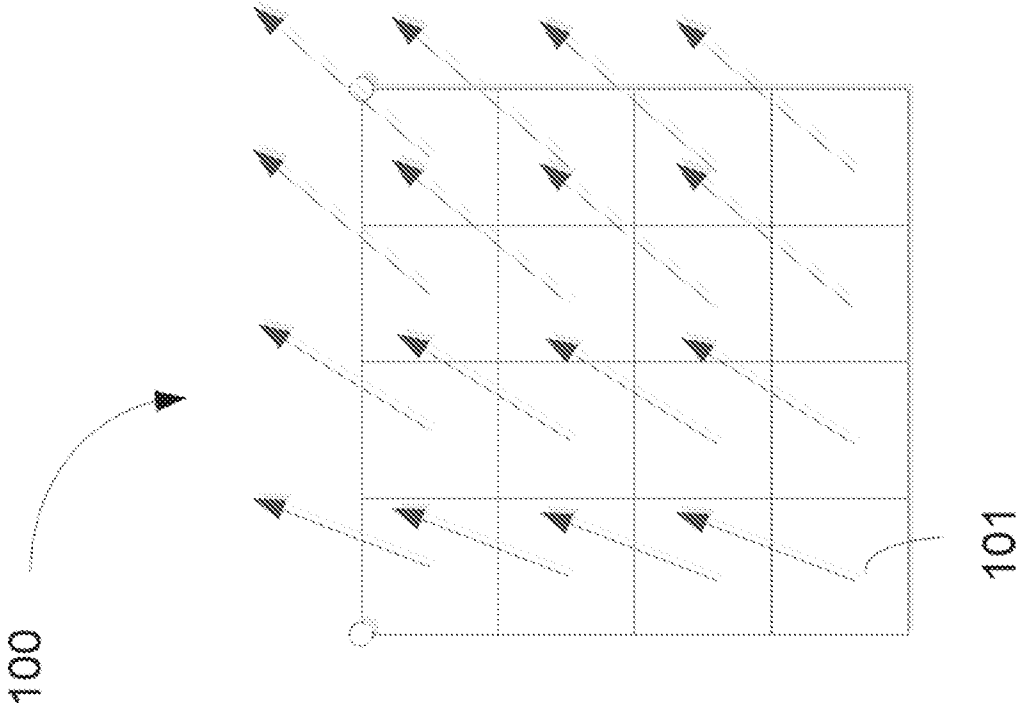


FIG. 1

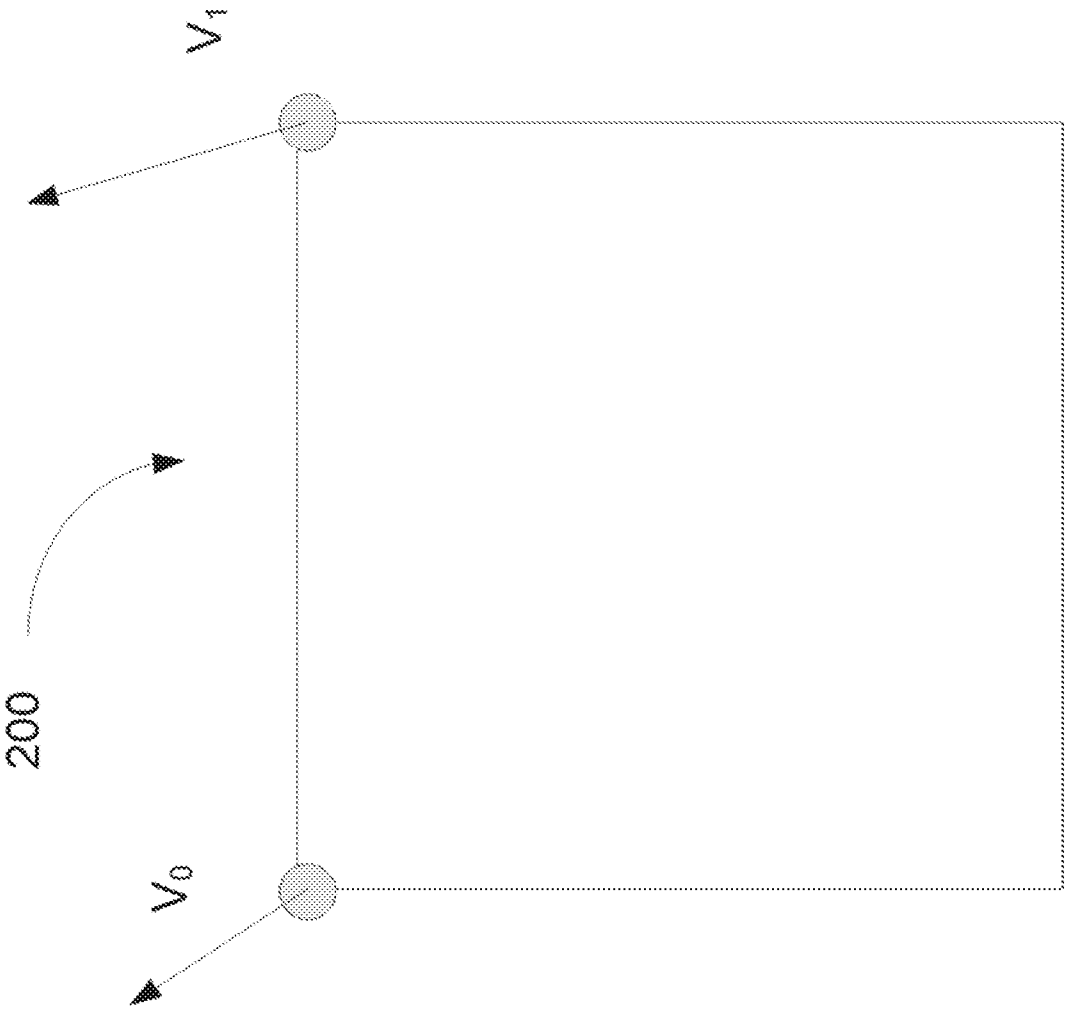


FIG. 2

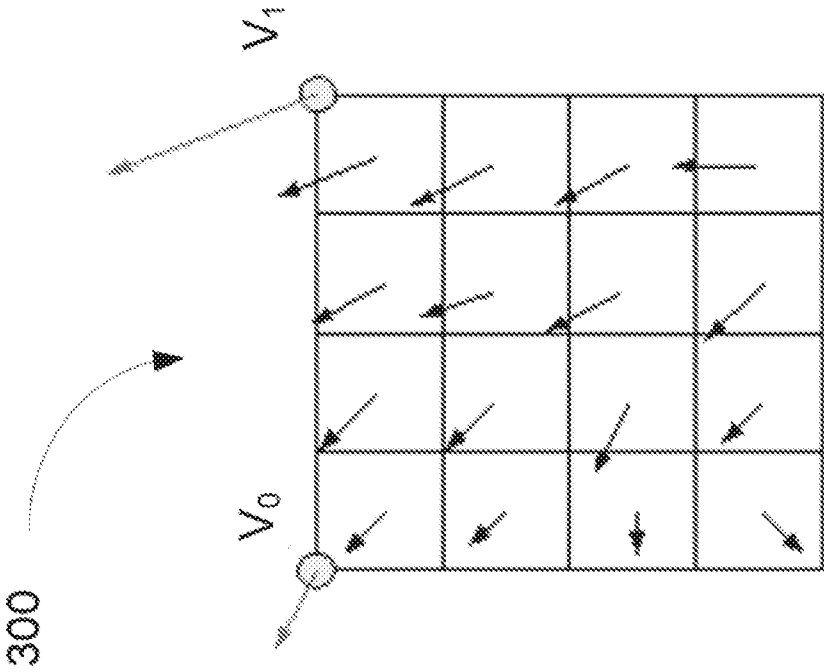


FIG. 3

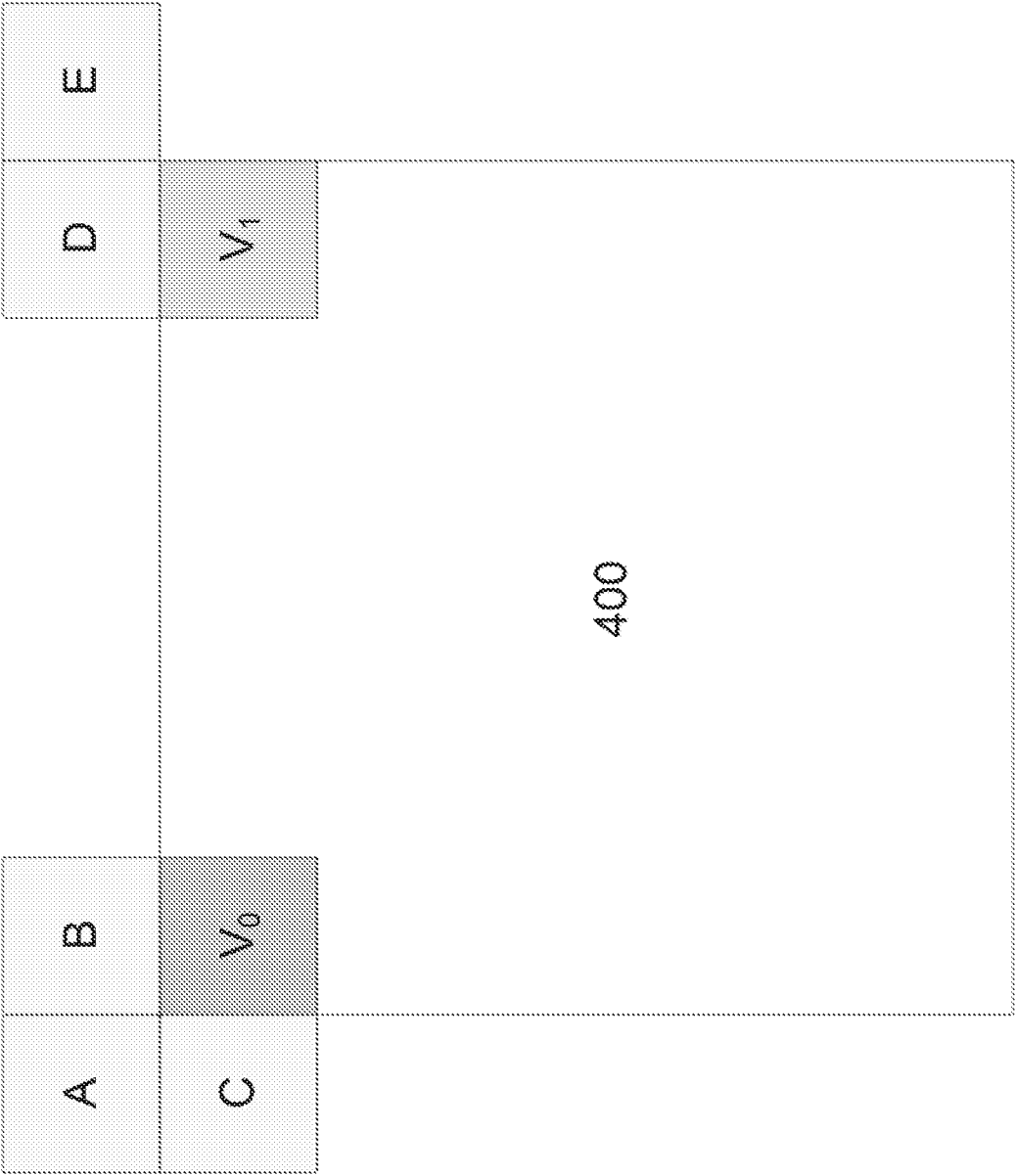


FIG. 4

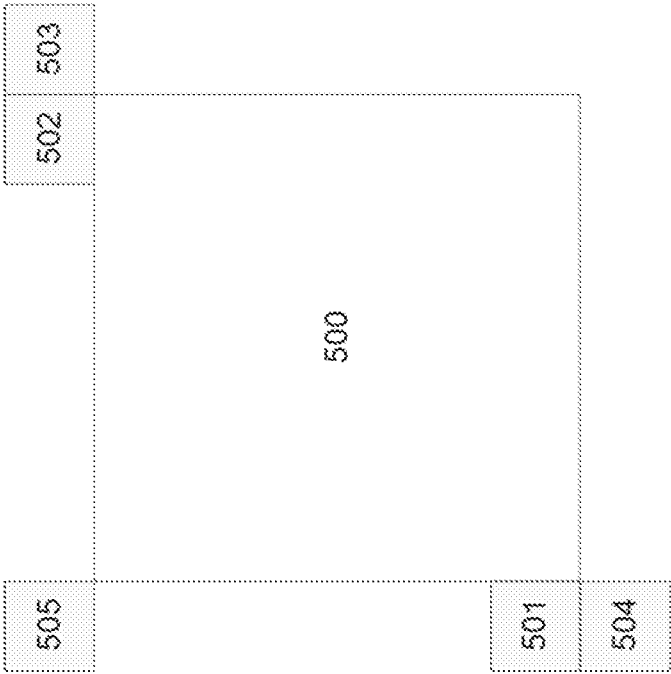


FIG. 5A

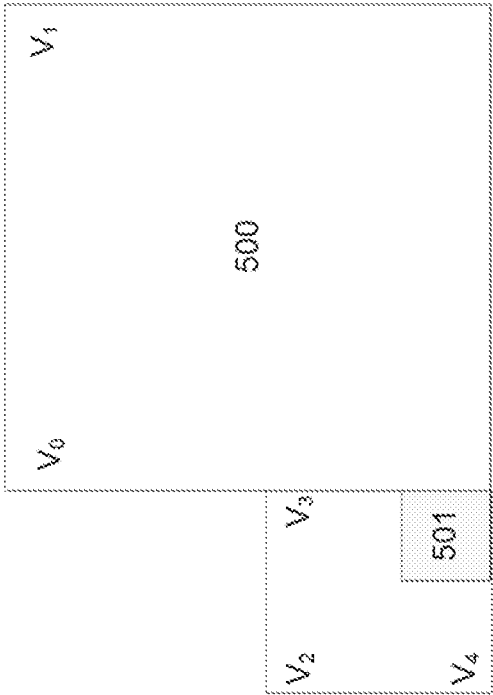


FIG. 5B

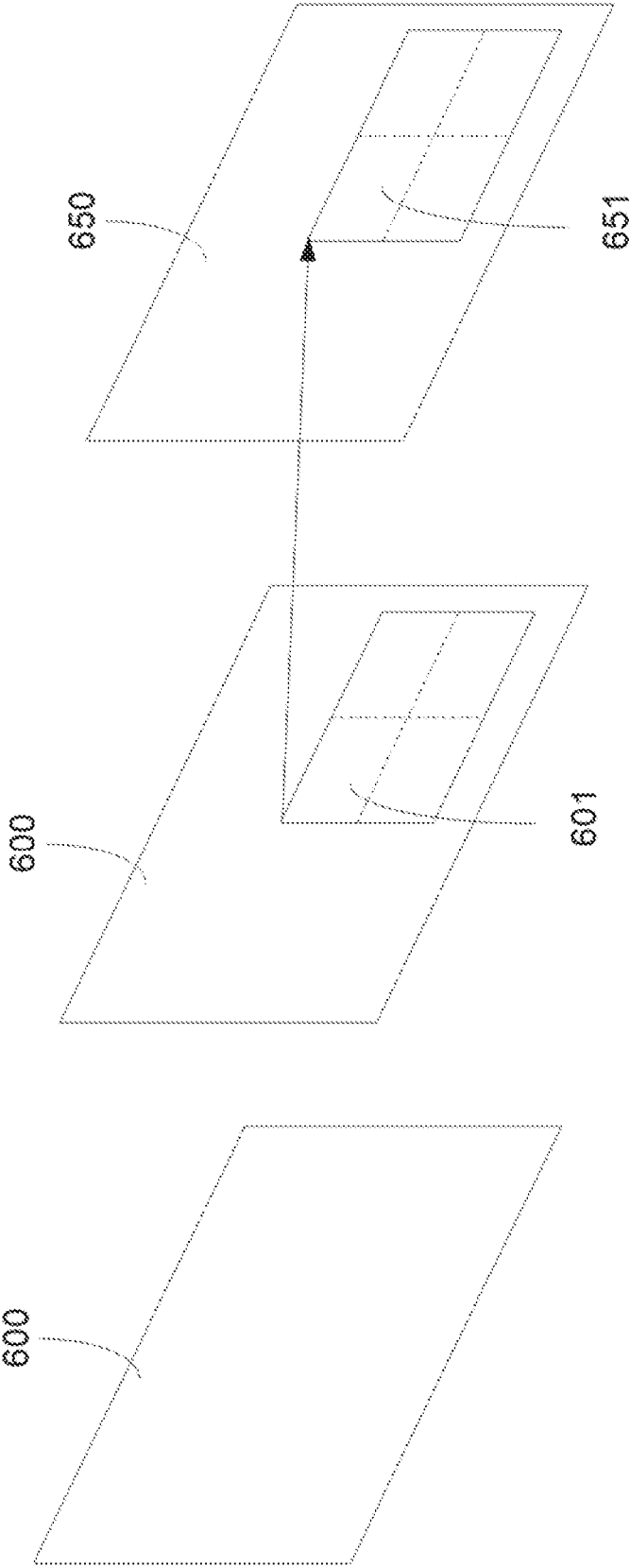


FIG. 6



FIG. 7

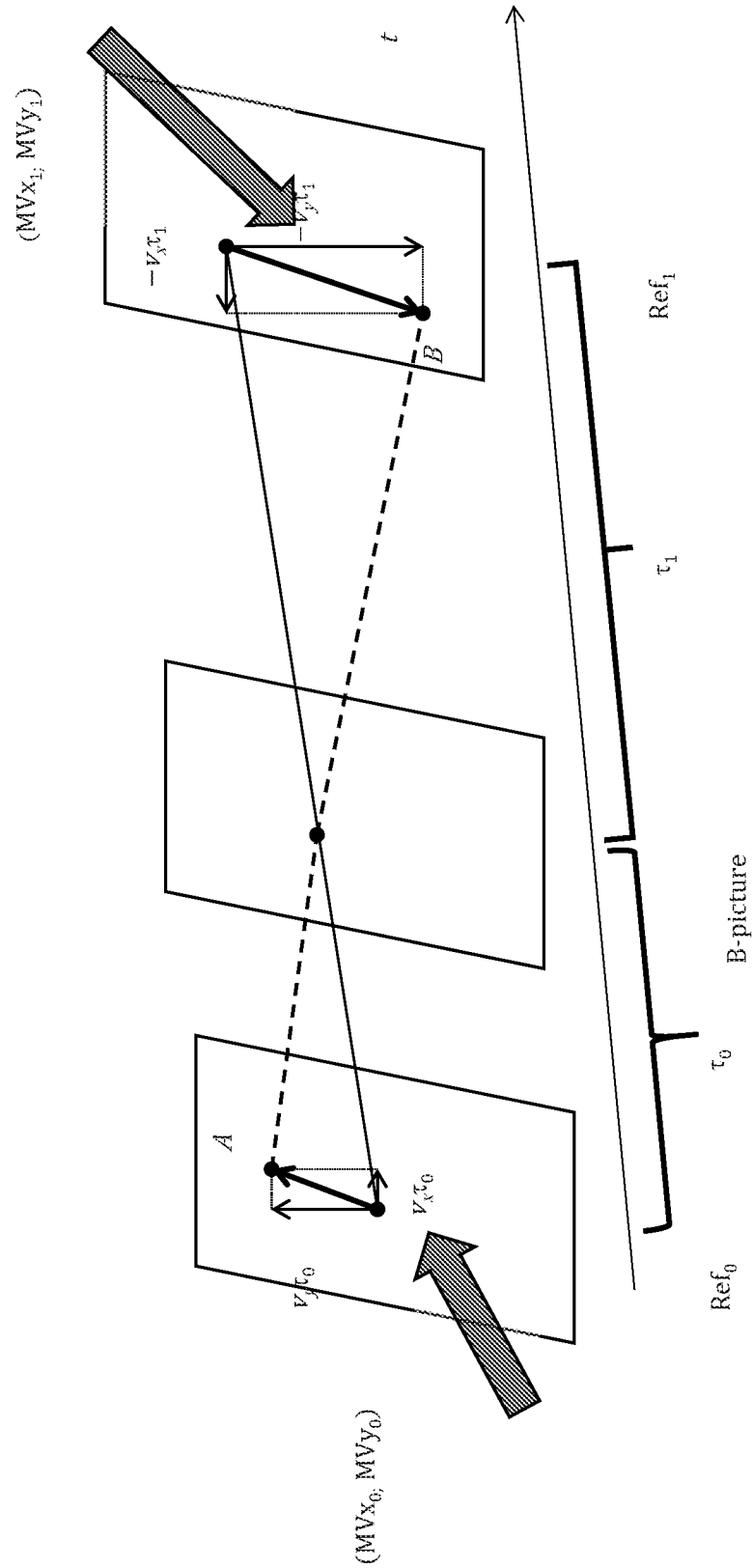


FIG. 8

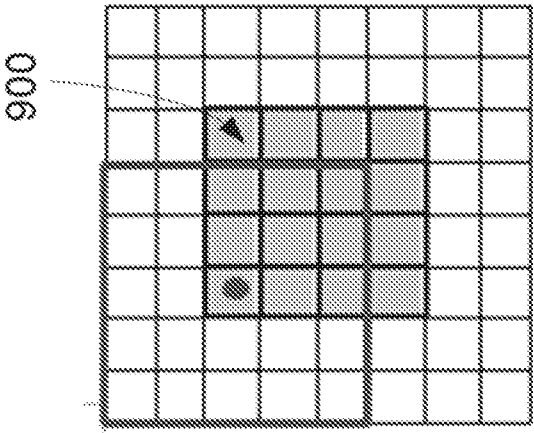


FIG. 9A

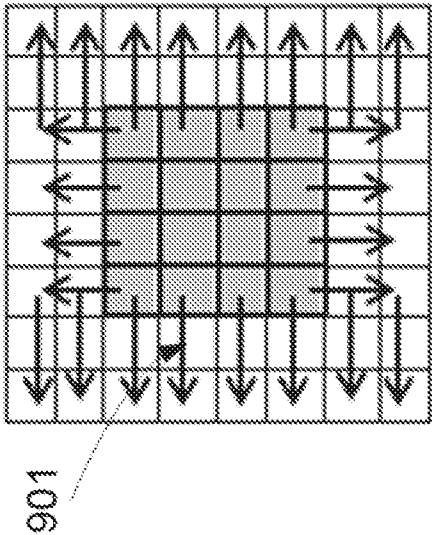


FIG. 9B

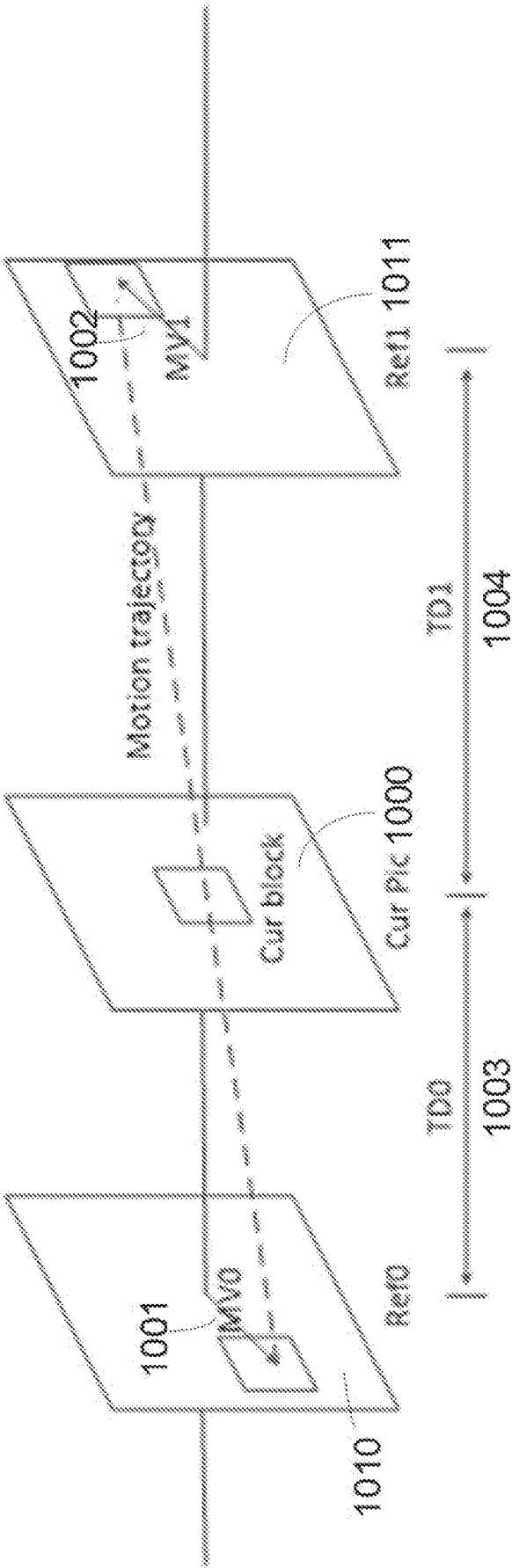


FIG. 10

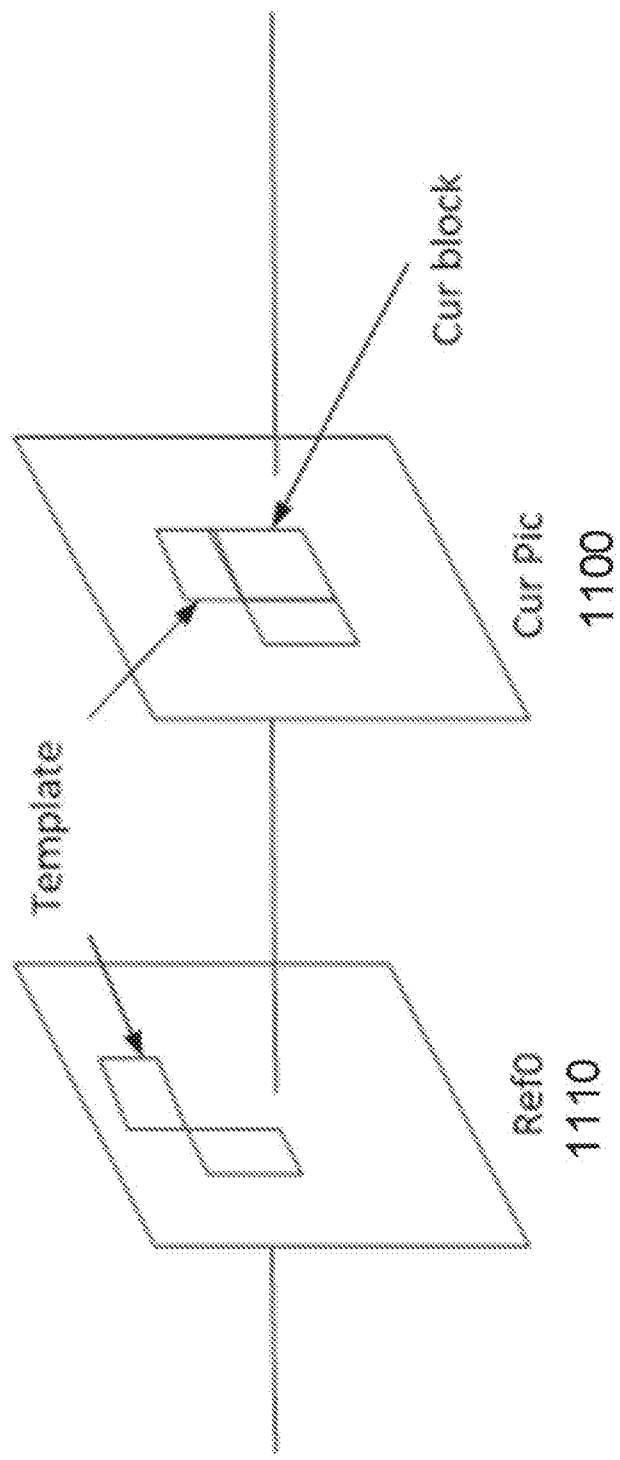


FIG. 11

1200

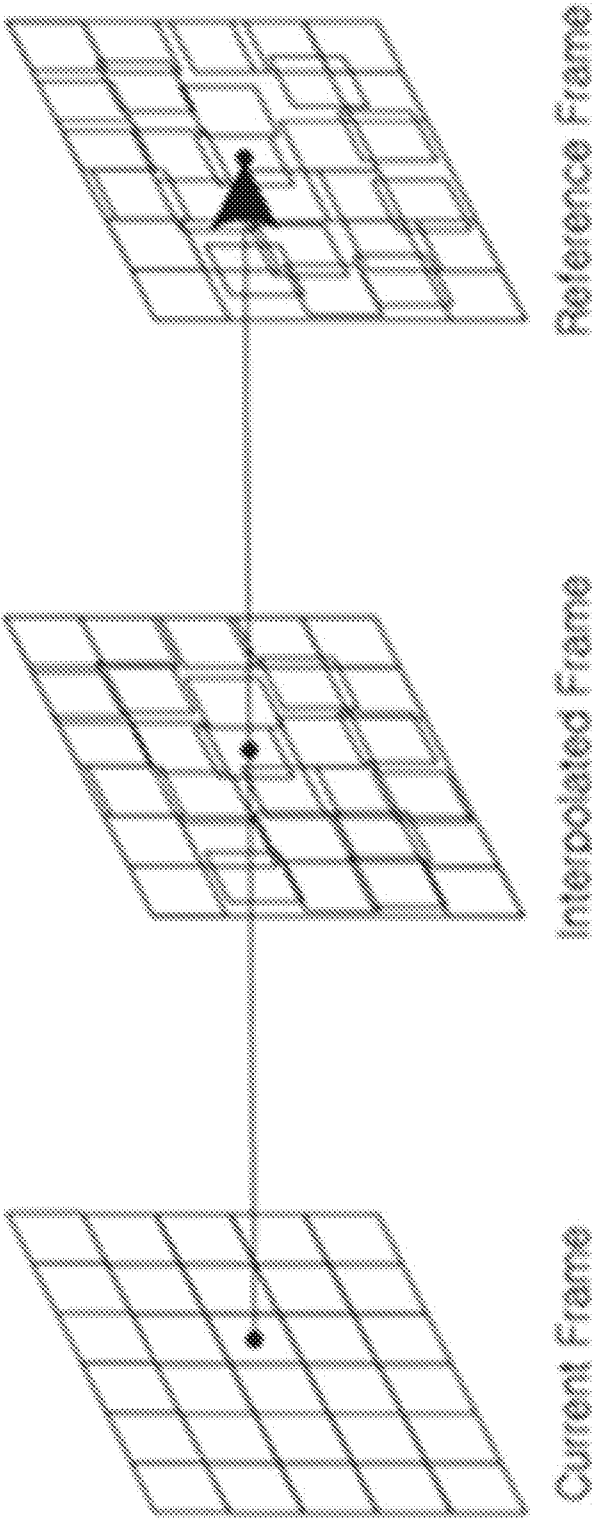


FIG. 12

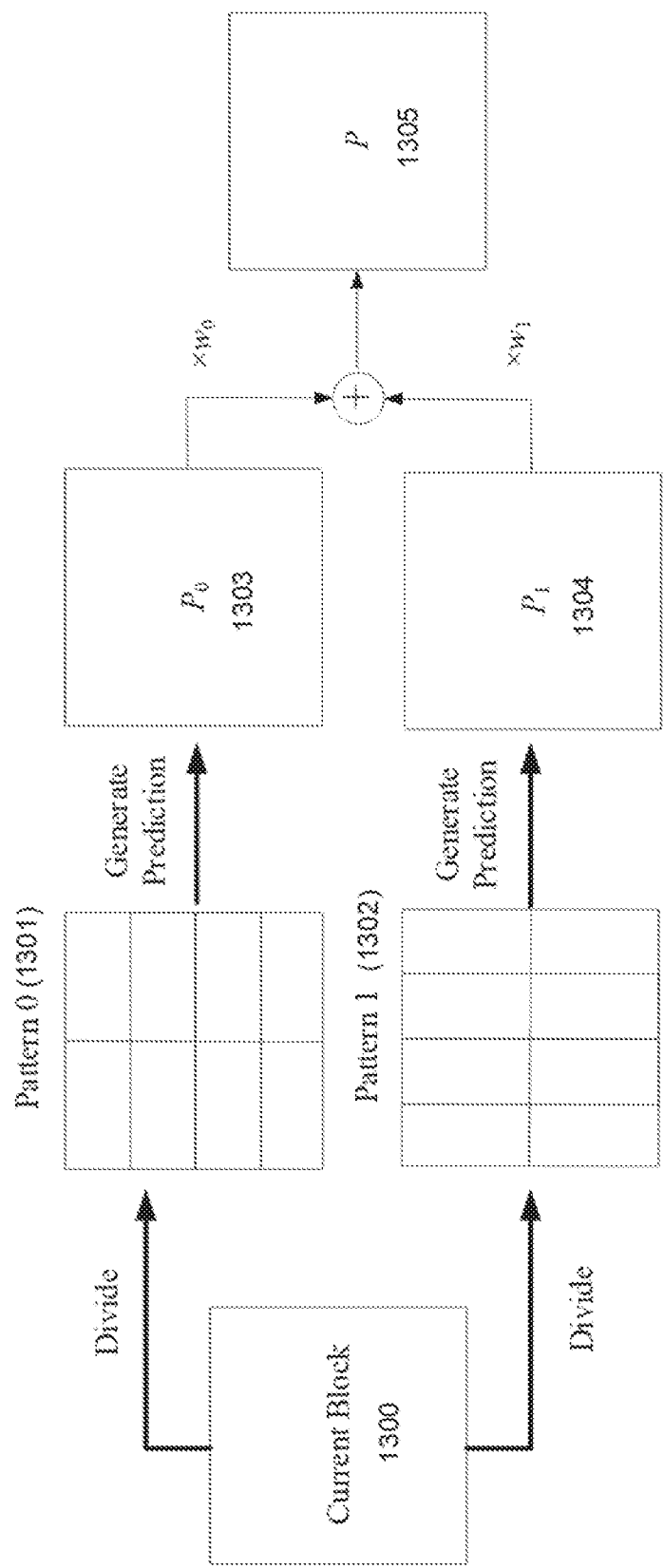


FIG. 13

Pattern 1

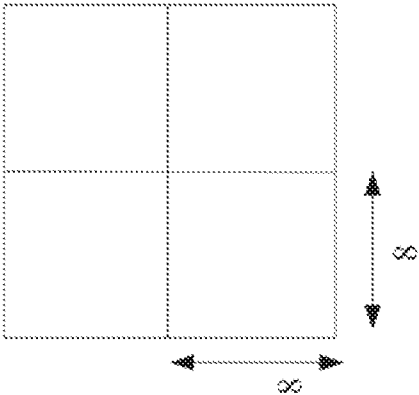


FIG. 14B

Pattern 0

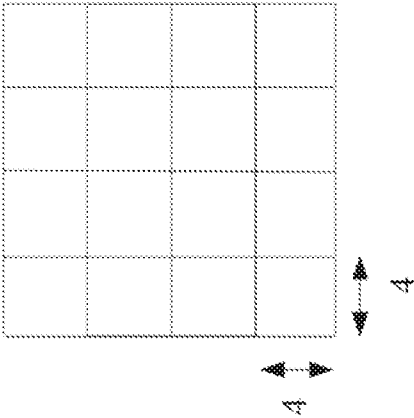


FIG. 14A

Pattern 3

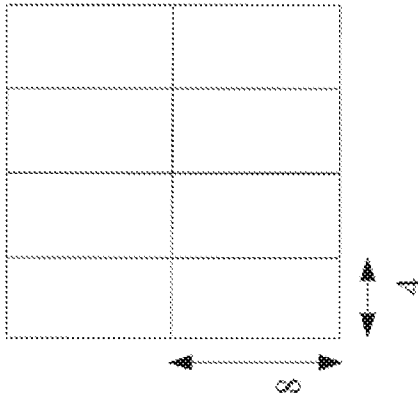


FIG. 14D

Pattern 2

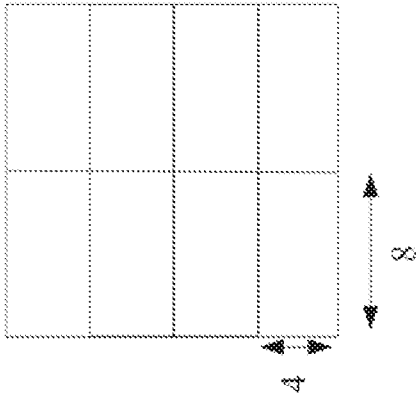


FIG. 14C

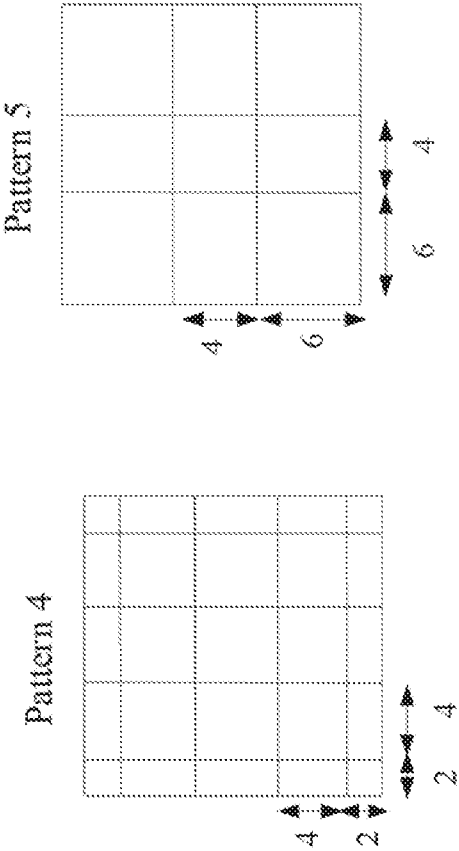


FIG. 14E

FIG. 14F

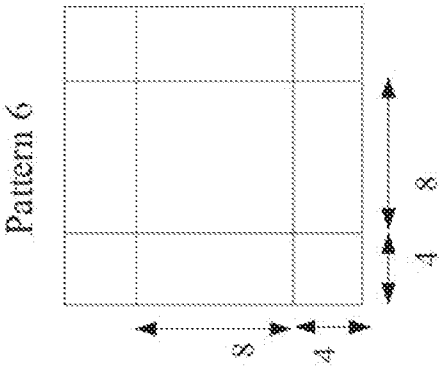


FIG. 14G

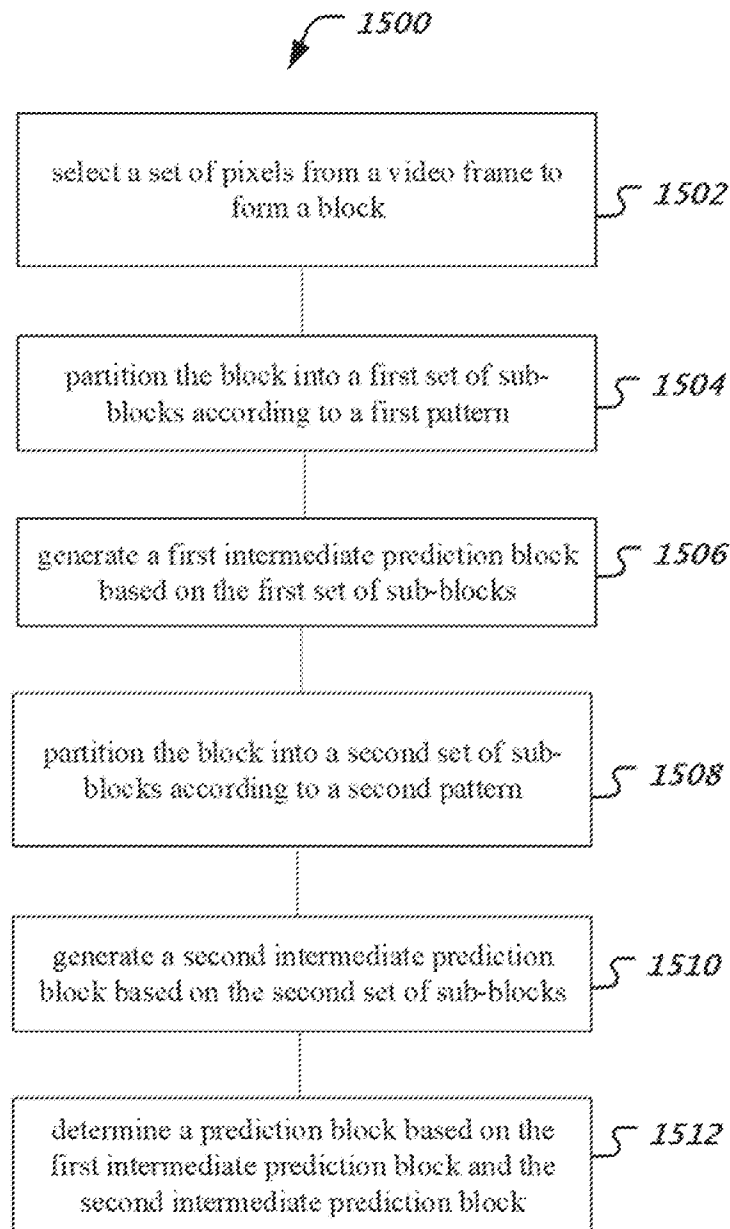


FIG. 15A

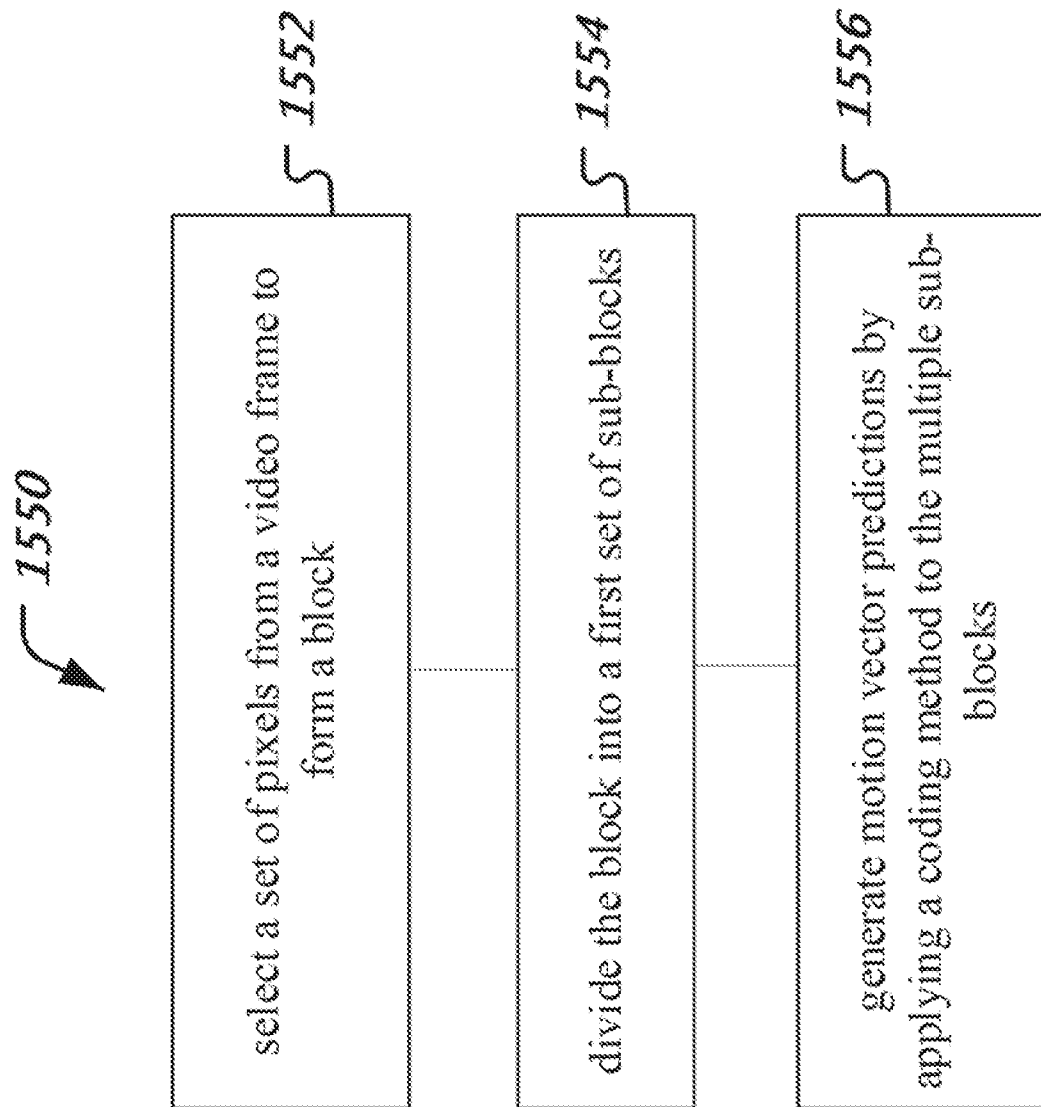


FIG. 15B

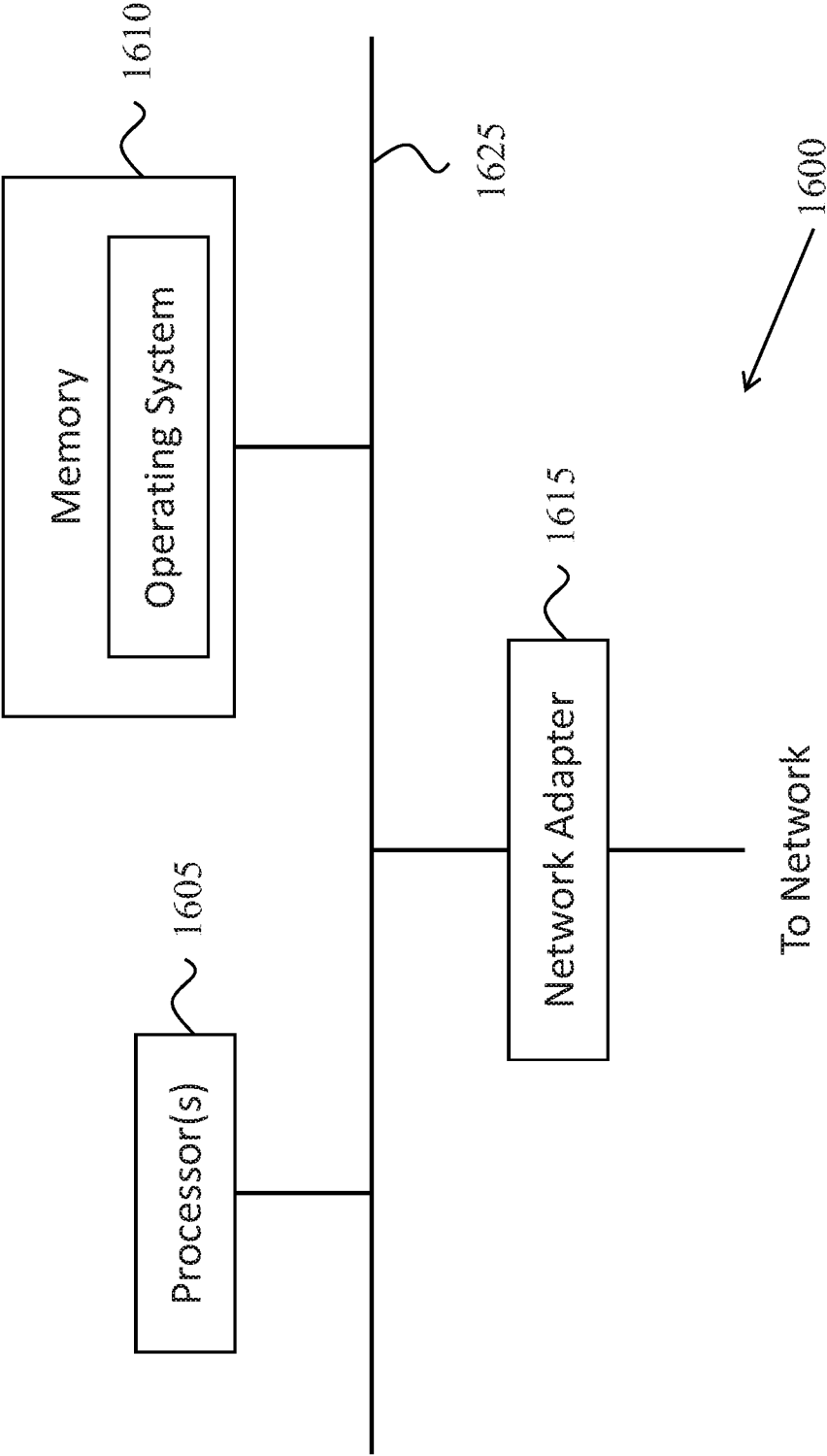


FIG. 16

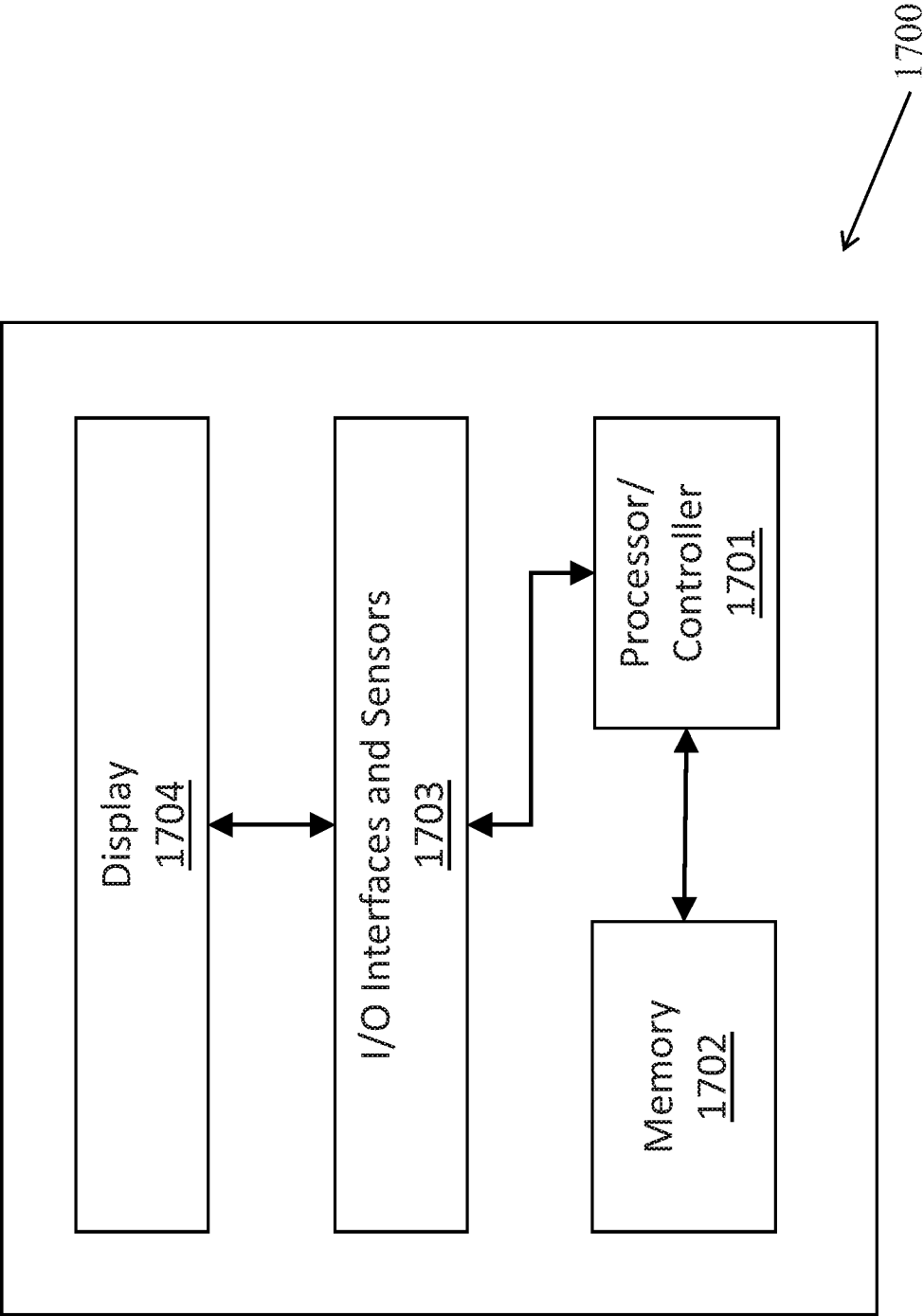


FIG. 17

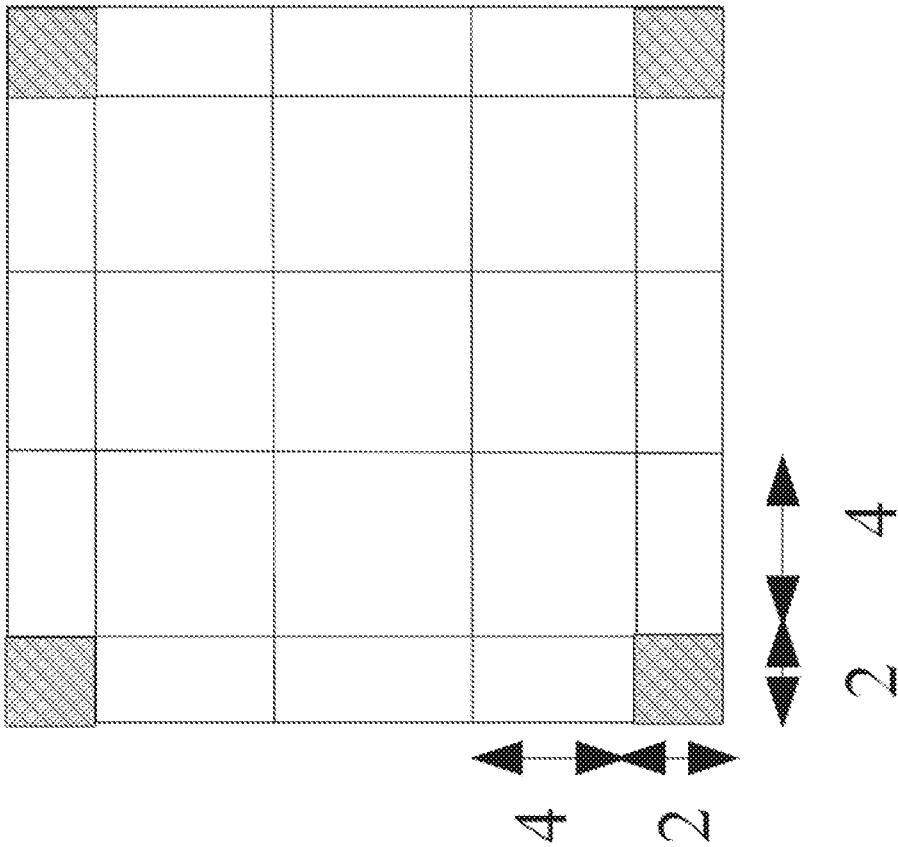


FIG. 18A

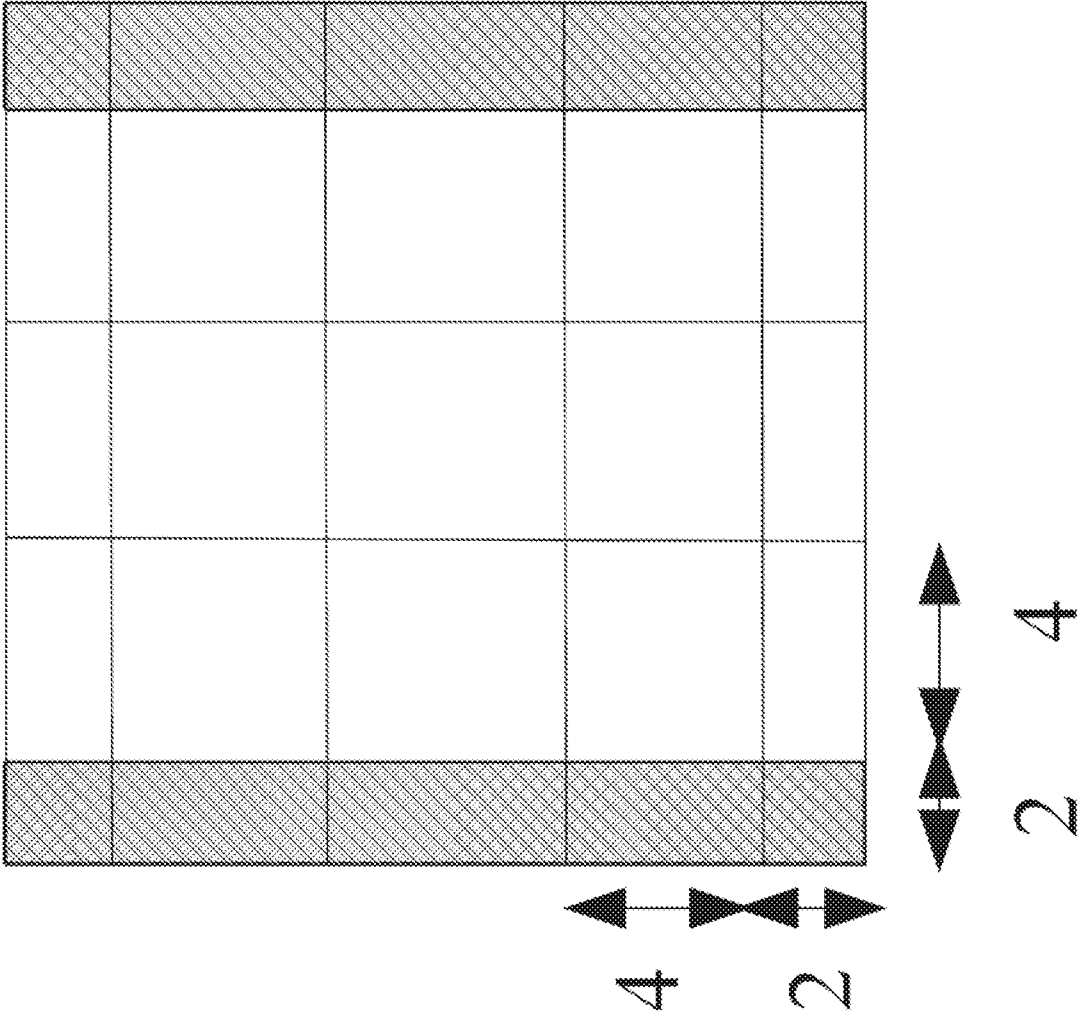


FIG. 18B

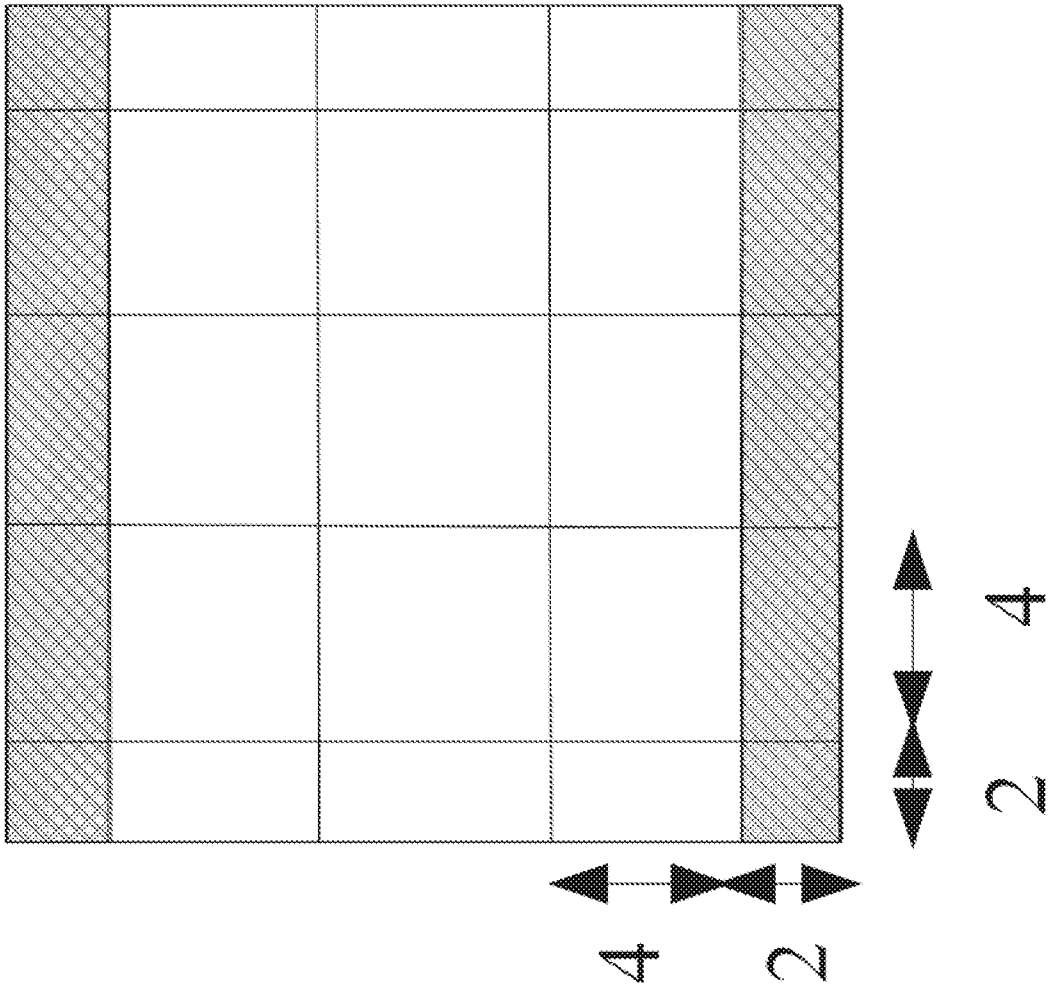


FIG. 18C

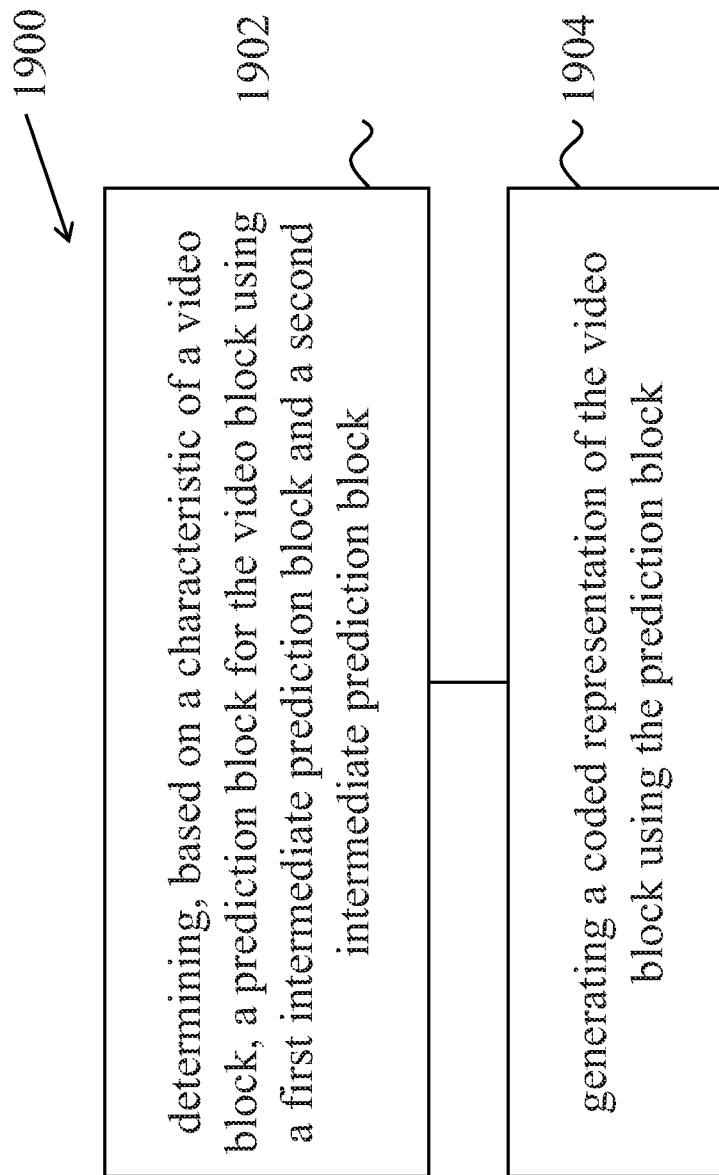


FIG. 19

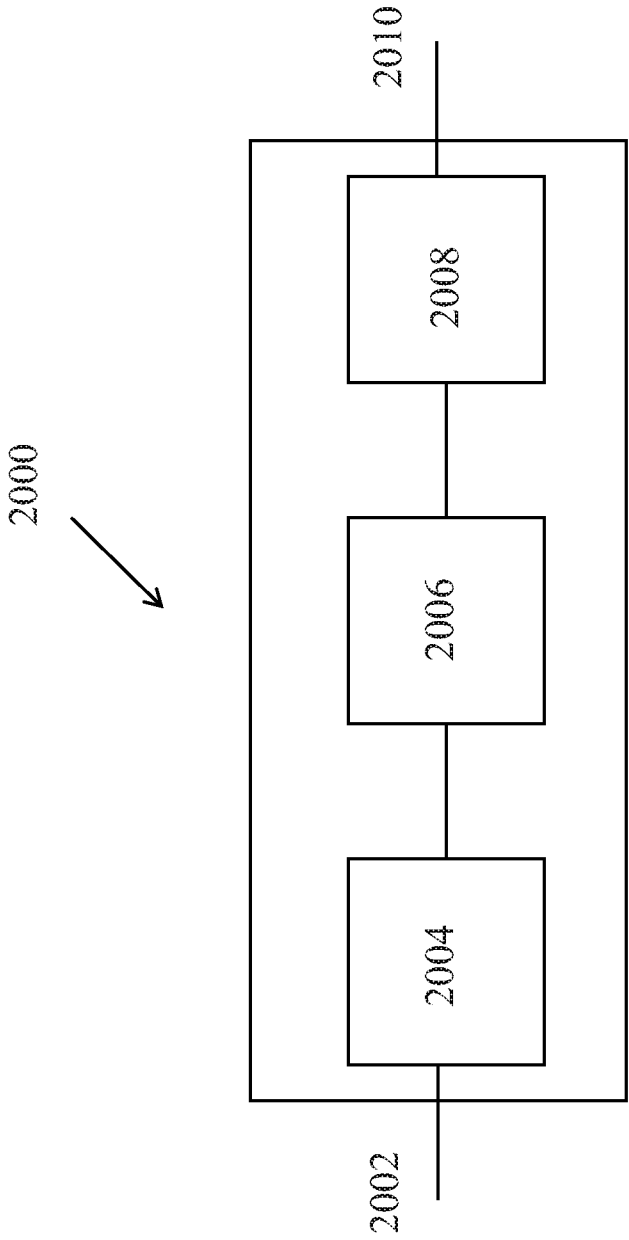


FIG. 20

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2019/057400

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04N19/105 H04N19/159 H04N19/176 H04N19/537
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)

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	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 11 July 2018 (2018-07-11), XP030199405, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0102-v2.zip JVET-K0102.pdf [retrieved on 2018-07-11]	1-3,15, 16
Y	the whole document ----- -/--	4-14



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

23 October 2019

Date of mailing of the international search report

06/11/2019

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

Bergmann, A

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2019/057400

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	ZHANG (BYTEDANCE) K ET AL: "CE4-related: Simplified Affine Prediction", 11. JVET MEETING; 20180711 - 20180718; LJUBLJANA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-K0103 11 July 2018 (2018-07-11), XP030199408, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end _user/documents/11_Ljubljana/wg11/JVET-K01 03-v2.zip JVET-K0103 r1.docx [retrieved on 2018-07-11] page 1 - page 2 -----	4-14