



(51) International Patent Classification:

H04N 19/573 (2014.01) *H04N 19/159* (2014.01)
H04N 19/105 (2014.01) *H04N 19/176* (2014.01)
H04N 19/139 (2014.01)

(21) International Application Number:

PCT/IB2019/055548

(22) International Filing Date:

01 July 2019 (01.07.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2018/093633

29 June 2018 (29.06.2018)

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,

(54) Title: BOUNDARY ENHANCEMENT FOR SUB-BLOCK

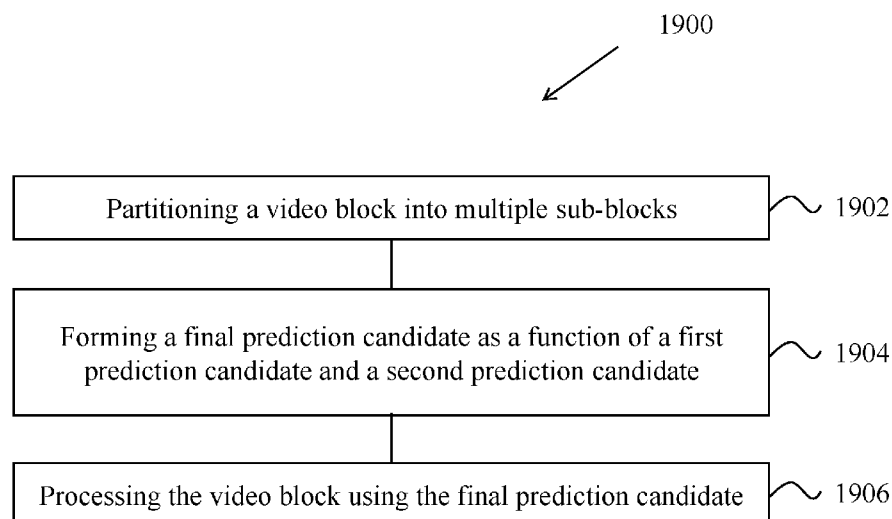


FIG. 19

(57) Abstract: Devices, systems and methods for boundary enhancement for sub-block based prediction are described. Partitioning a block can result in discontinuities between adjacent sub-blocks that may introduce some undesirable high-frequency energy in the residual signal, which can deteriorate the performance of subsequent transform coding. Implementations of the disclosed technology can reduce the effect of the discontinuities. In a representative aspect, a method for video coding includes partitioning a video block into multiple sub-blocks, forming a first prediction candidate based on a sub-block based prediction of a plurality of samples in the video block, forming a second prediction candidate based on an inter prediction of the plurality of samples in a sub-block boundary region, forming a final prediction candidate as a function of the first prediction candidate and the second prediction candidate, and processing the video block using the final prediction candidate.

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, 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*

BOUNDARY ENHANCEMENT FOR SUB-BLOCK

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/CN2018/093633, filed on June 29, 2018. For all purposes under the U.S. law, the entire disclosure of the International Patent Application No. PCT/CN2018/093633 is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[0002] This patent document is directed generally to image and video coding technologies.

BACKGROUND

[0003] Digital video accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

[0004] Motion compensation 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 and decoding of video data for video compression.

SUMMARY

[0005] Devices, systems, and methods related to boundary enhancement for sub-block based prediction for image and video coding are described.

[0006] In one representative aspect, the disclosed technology may be used to provide a method for video coding. This method includes partitioning a video block into multiple sub-blocks, forming a first prediction candidate based on a sub-block based prediction of a plurality of samples in the video block, forming a second prediction candidate based on an inter prediction of the plurality of samples in a sub-block boundary region, forming a final prediction candidate as a function of the first prediction candidate and the second prediction candidate, and processing

the video block using the final prediction candidate.

[0007] In another representative aspect, the above-described method is embodied in the form of processor-executable code and stored in a computer-readable program medium.

[0008] In yet another representative aspect, a device that is configured or operable to perform the above-described method is disclosed. The device may include a processor that is programmed to implement this method.

[0009] In yet another representative aspect, a video decoder apparatus may implement a method as described herein.

[0010] The above and other aspects and features of the disclosed technology are described in greater detail in the drawings, the description, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 shows an example of sub-block based prediction.

[0012] FIG. 2 shows an example of a simplified affine motion model.

[0013] FIG. 3 shows an example of an affine motion vector field (MVF) per sub-block.

[0014] FIG. 4 shows an example of motion vector prediction (MVP) for the AF_INTER affine motion mode.

[0015] FIGS. 5A and 5B show example candidates for the AF_MERGE affine motion mode.

[0016] FIG. 6 shows an example of motion prediction using the alternative temporal motion vector prediction (ATMVP) algorithm for a coding unit (CU).

[0017] FIG. 7 shows an example of a coding unit (CU) with sub-blocks and neighboring blocks used by the spatial-temporal motion vector prediction (STMVP) algorithm.

[0018] FIG. 8 shows an example of an optical flow trajectory used by the bi-directional optical flow (BIO) algorithm.

[0019] FIGS. 9A and 9B show example snapshots of using of the bi-directional optical flow (BIO) algorithm without block extensions.

[0020] FIG. 10 shows an example of bilateral matching in the frame-rate up conversion (FRUC) algorithm.

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

[0022] FIG. 12 shows an example of unilateral motion estimation in the FRUC algorithm.

[0023] FIGS. 13A and 13B show examples of boundaries of sub-blocks that may be filtered.

[0024] FIGS. 14A and 14B show examples of prediction samples to be filtered.

[0025] FIGS. 15A, 15B, 15C, and 15D show examples of boundary enhancement for affine prediction, in accordance with the disclosed technology.

[0026] FIG. 16 shows a flowchart of an example method for video coding in accordance with the disclosed technology.

[0027] FIG. 17 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.

[0028] FIG. 18 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.

[0029] FIG. 19 is a flowchart for an example method of video processing.

DETAILED DESCRIPTION

[0030] Due to the increasing demand for higher resolution video, video coding methods and techniques are ubiquitous in modern technology. Video codecs typically include an electronic circuit or software that compresses or decompresses digital video, and are continually being improved to provide higher coding efficiency. A video codec converts uncompressed video to a compressed format or vice versa. There are complex relationships between the video quality, the amount of data used to represent the video (determined by the bit rate), the complexity of the encoding and decoding algorithms, sensitivity to data losses and errors, ease of editing, random access, and end-to-end delay (latency). The compressed format usually conforms to a standard video compression specification, e.g., the High-Efficiency Video Coding (HEVC) standard (also known as H.265 or MPEG-H Part 2), the Versatile Video Coding standard to be finalized, or other current and/or future video coding standards.

[0031] Sub-block based prediction is first introduced into the video coding standard by the High-Efficiency Video Coding (HEVC) standard. With sub-block based prediction, a block, such as a Coding Unit (CU) or a Prediction Unit (PU), is divided into several non-overlapped sub-blocks. Different sub-blocks may be assigned different motion information, such as reference index or motion vector (MV), and motion compensation (MC) is performed individually for each sub-block. FIG. 1 shows an example of sub-block based prediction.

[0032] Embodiments of the disclosed technology may be applied to existing video coding

standards (e.g., HEVC, H.265) and future standards to improve runtime performance. Section headings are used in the present document to improve readability of the description and do not in any way limit the discussion or the embodiments (and/or implementations) to the respective sections only.

[0033] Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Examples of the Joint Exploration Model (JEM)

[0034] In some embodiments, future video coding technologies are explored using a reference software known as the Joint Exploration Model (JEM). In JEM, sub-block based prediction is adopted in several coding tools, such as affine prediction, alternative temporal motion vector prediction (ATMVP), spatial-temporal motion vector prediction (STMVP), bi-directional optical flow (BIO), Frame-Rate Up Conversion (FRUC), Locally Adaptive Motion Vector Resolution (LAMVR), Overlapped Block Motion Compensation (OBMC), Local Illumination Compensation (LIC), and Decoder-side Motion Vector Refinement (DMVR).

1.1 Examples of affine prediction

[0035] In HEVC, only a 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:

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

[0037] 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:

$$[0038] \quad \begin{cases} M = \text{clip3}(4, w, \frac{w \times MvPre}{\max(\text{abs}(v_{1x} - v_{0x}), \text{abs}(v_{1y} - v_{0y}))}) \\ N = \text{clip3}(4, h, \frac{h \times MvPre}{\max(\text{abs}(v_{2x} - v_{0x}), \text{abs}(v_{2y} - v_{0y}))}) \end{cases} \quad \text{Eq. (2)}$$

[0039] Here, $MvPre$ is the motion vector fraction accuracy (e.g., 1/16 in JEM). (v_{2x}, v_{2y}) is the motion vector of the bottom-left control point, calculated according to Eq. (1). M and N can be adjusted downward if necessary to make it a divisor of w and h , respectively.

[0040] FIG. 3 shows an example of affine MVF per sub-block for a block 300. To derive the 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.

[0041] 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.

[0042] 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.

[0043] 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.

[0044] 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.

[0045] In JEM, the non-merge affine mode can be used only when the width and the height of the current block are both larger than 8; the merge affine mode can be used only when the area (i.e., width $\times$ height) of the current block is not smaller than 64.

1.2 Examples of alternative temporal motion vector prediction (ATMVP)

[0046] 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.

[0047] 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.

[0048] 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.

[0049] 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.

1.3 Examples of spatial-temporal motion vector prediction (STMVP)

[0050] In the STMVP method, the motion vectors of the sub-CUs are derived recursively, following a 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).

[0051] 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.

1.4 Examples of bi-directional optical flow (BIO)

[0052] The bi-directional optical flow (BIO) method is a 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.

[0053] 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:

$$\mathbf{[0054]} \quad \partial I^{(k)}/\partial t + v_x \partial I^{(k)}/\partial x + v_y \partial I^{(k)}/\partial y = 0. \quad \text{Eq. (3)}$$

[0055] 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 $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ derivatives at the ends. The value of this polynomial at $t=0$ is the BIO prediction:

$$\mathbf{[0056]} \quad pred_{BIO} = 1/2 \cdot (I^{(0)} + I^{(1)} + v_x/2 \cdot (\tau_1 \partial I^{(1)}/\partial x - \tau_0 \partial I^{(0)}/\partial x) + v_y/2 \cdot (\tau_1 \partial I^{(1)}/\partial y - \tau_0 \partial I^{(0)}/\partial y)) \quad \text{Eq. (4)}$$

[0057] 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., $MVx_0, MVy_0, MVx_1, MVy_1 \neq 0$) and the block motion vectors are proportional to the time distance (e.g., $MVx_0/MVx_1 = MVy_0/MVy_1 = -\tau_0/\tau_1$).

[0058] 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 :

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

[0060] 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:

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

[0062] 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:

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

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

[0065] where,

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

[0067] 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), where:

$$[0068] \quad r = 500 \cdot 4^{d-8} \quad \text{Eq. (10)}$$

$$[0069] \quad m = 700 \cdot 4^{d-8} \quad \text{Eq. (11)}$$

[0070] Here, d is bit depth of the video samples.

[0071] 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 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.

[0072] 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 4×4 block. In the block-based BIO, the values of s_n in Eq. (9) of all samples in a 4×4 block can be aggregated, and then the aggregated values of s_n are used to derived BIO motion vectors offset for the 4×4 block. More specifically, the following formula can used for block-based BIO derivation:

$$[0073] \quad \begin{aligned} s_{1,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega(x,y)} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_{3,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) \\ s_{2,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y); \\ s_{5,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_{6,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad \text{Eq. (12)}$$

[0074] Here, b_k denotes the set of samples belonging to the k -th 4×4 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.

[0075] 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 $12 \times 2^{13-d}$.

[0076] 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 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 2 shows example filters that can be used for gradients calculation of different fractional positions of block motion vector in BIO. Table 3 shows example interpolation filters that can be used for prediction signal generation in BIO.

Table 2: Example filters for gradient calculation in BIO

Fractional pel position	Interpolation filter for gradient(<i>BIOfilterG</i>)
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 3: 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 }

[0077] 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.

[0078] 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.

[0079] In JEM, BIO is only invoked for the luma component.

1.5 Examples of frame-rate up conversion (FRUC)

[0080] 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.

[0081] At the 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.

[0082] Typically, motion derivation process in FRUC merge mode has two steps: a CU-level

motion search is first performed, then followed by a Sub-CU level motion refinement. At CU level, an initial motion vector is derived for the whole CU based on bilateral matching or template matching. First, a list of MV candidates is generated, and the candidate that leads to the minimum matching cost is selected as the starting point for further CU level refinement. Then a local search based on bilateral matching or template matching around the starting point is performed. The MV results in the minimum matching cost are 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.

[0083] For example, the following derivation process is performed for a $W \times H$ CU motion information derivation. At the first stage, MV for the whole $W \times H$ CU is derived. At the second stage, the CU is further split into $M \times M$ sub-CUs. The value of M is calculated as in (16), D is a predefined splitting depth which is set to 3 by default in the JEM. Then the MV for each sub-CU is derived.

$$\mathbf{[0084]} \quad M = \max\{4, \min\{\frac{M}{2^D}, \frac{N}{2^D}\}\} \quad \text{Eq. (13)}$$

[0085] FIG. 10 shows an example of bilateral matching used in the Frame-Rate Up Conversion (FRUC) method. The bilateral matching is used to derive motion information of the current CU by finding the closest match between two blocks along the motion trajectory of the current CU (1000) in two different reference pictures (1010, 1011). Under the assumption of continuous motion trajectory, the motion vectors MV0 (1001) and MV1 (1002) pointing to the two reference blocks are proportional to the temporal distances, e.g., TD0 (1003) and TD1 (1004), between the current picture and the two reference pictures. In some embodiments, when the current picture 1000 is temporally between the two reference pictures (1010, 1011) and the temporal distance from the current picture to the two reference pictures is the same, the bilateral matching becomes mirror based bi-directional MV.

[0086] FIG. 11 shows an example of template matching used in the Frame-Rate Up Conversion (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.

[0087] 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.

[0088] 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 .

[0089] 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.

[0090] 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.

[0091] Generation of an interpolated MV field. 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.

[0092] 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.

[0093] Interpolation and matching cost. 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.

[0094] 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:

$$[0095] \quad C = SAD + w \cdot (|MV_x - MV_x^s| + |MV_y - MV_y^s|) \quad \text{Eq. (14)}$$

[0096] 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.

[0097] 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.

[0098] 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.

[0099] 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:

[00100] If $\text{costBi} \leq \text{factor} * \min(\text{cost0}, \text{cost1})$

[00101] bi-prediction is used;

[00102] Otherwise, if $\text{cost0} \leq \text{cost1}$

[00103] uni-prediction from list0 is used;

[00104] Otherwise,

[00105] uni-prediction from list1 is used;

[00106] 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.

1.6 Examples of MV derived for MC in chroma components

[00107] In an example, the HEVC standard defines how to derive the MV used for MC in chroma components (noted as mvC) from the MV used for MC in the luma component (noted as mv). Generally speaking, mvC is calculated as mv multiplying a factor, which relies on the color format, such as 4:2:0 or 4:2:2.

2. Examples of existing methods for sub-block based implementation

[00108] In some existing implementations, sub-block based prediction is used since it is usually more accurate than the whole block prediction because it can partition a block into more parts with their own MVs.

[00109] However, the partitioning may result in discontinuities between two adjacent sub-

blocks along their boundary. The discontinuities may introduce some undesirable high-frequency energy in the residual signal, which can deteriorate the performance of subsequent transform coding.

3. Exemplary methods for sub-block based prediction in video coding

[00110] The use of boundary enhancement for sub-block based prediction to improve video coding efficiency and enhance both existing and future video coding standards is elucidated in the following examples described for various implementations. In the following examples, which should not be construed to be limiting, the width and height of the current block for a component are noted as W and H respectively, the width and height of the sub-block assigned to the component are noted as w and h respectively.

[00111] Example 1. The samples along the boundaries of sub-blocks are predicted by the sub-block based prediction (named the first prediction); they are also predicted by another inter-prediction (named the second prediction). The first prediction and the second prediction are used jointly to derive the final prediction for a sample along the boundary. The final prediction for a sample not along the boundaries may not be changed, e.g., equal to the first prediction.

[00112] (a) In one example, the boundaries only include the inner boundaries, e.g., the boundaries between sub-blocks, as shown in the example in FIG. 13A. The shaded regions cover the samples along the boundaries.

[00113] (b) In one example, the boundaries include both the inner boundaries and the outer boundaries, e.g., the boundaries between sub-blocks and other blocks already coded or decoded, as shown in the example in FIG. 13B. The shaded regions cover the samples along the boundaries.

[00114] Example 2. In one example, the range of samples along the boundaries to be enhanced can be predefined or adaptive. For example, there can be N columns of samples along a vertical boundary and M rows of samples along a horizontal boundary to be enhanced. FIGS. 14A and 14B show examples of prediction samples for $M=N=2$.

[00115] (a) In one example, M and/or N depend on the width and height of the sub-block. For example, $M=N=2$ if the sub-block's shape is 4×4 ; $M=N=4$ if the sub-block's shape is 8×8 .

[00116] (b) In one example, M and/or N depend on color component. For example, $M=N=2$ for the luma component; and $M=N=1$ for the chroma components.

[00117] (c) In one example, M and/or N depend on the location of the boundary. For example, $M=N=2$ if the boundary is between the sub-block and a coded/decoded neighboring block; and $M=N=1$ if the boundary is between two sub-blocks.

[00118] (d) In one example, M and/or N may depend on the location of the sub-block. Alternatively, it may depend on how many neighboring blocks are coded/decoded and/or how many prediction blocks of neighboring blocks are available.

[00119] (e) In one example, M and/or N are signaled from the encoder to the decoder. For example, M and N can be signaled in Video Parameter Set (VPS), Sequence Parameter Set (SPS), Picture Parameter Set (PPS), Slice header, Coding Tree Unit (CTU) or Coding Unit (CU).

[00120] Example 3. In one example, the first prediction could be obtained via the conventional way of the sub-block based prediction, while the second prediction is generated with the same model which generates the first prediction, but with different positions. For example, both the first prediction and second prediction are generated by eq. (1), but with different (x, y) s.

[00121] (a) In one example, the first prediction is generated with the affine prediction approach in JEM as shown in FIG. 15A, where the MVs for each sub-block is obtained by setting (x, y) in eq.(1) at the center of the sub-block. That is, suppose the top-left point of a sub-block is at $(i \times w, j \times h)$, then the MV for this sub-block (the (i, j) th sub-blocks) is calculated from eq.(1) with $(x, y) = (i \times w + w/2, j \times h + h/2)$. Then the second prediction can be generated as

[00122] (i) at vertical boundaries as shown in FIG. 15B. The second prediction is generated by MC using auxiliary sub-blocks covering the vertical boundaries. The size of auxiliary sub-blocks, noted as $w' \times h'$ can be the same as that of the original sub-blocks, e.g., $w' = w$ and $h' = h$. Or the two sizes can be different. In the example of FIG. 15B where $w' = w$ and $h' = h$, the top-left top of an auxiliary sub-block (at the i th row and j th column of the auxiliary sub-blocks) is $(i \times w + w/2, j \times h)$ and the MV for this auxiliary sub-block is calculated from eq.(1) with $(x, y) = (i \times w + w, j \times h + h/2)$.

[00123] (ii) at horizontal boundaries as shown in FIG. 15C. The second prediction is generated by MC using auxiliary sub-blocks covering the horizontal boundaries. The size of auxiliary sub-blocks, noted as $w' \times h'$ can be the same as that of the original sub-blocks, i.e., $w' = w$ and $h' = h$. Or the two sizes can be different. In the example of FIG. 15C where $w' = w$ and

$h'=h$, the top-left top of an auxiliary sub-block is $(i \times w, j \times h + h/2)$ and the MV for this auxiliary sub-block is calculated from eq.(1) with $(x,y) = (i \times w + w/2, j \times h + h)$.

[00124] (iii) at the cross-points of vertical boundaries and horizontal boundaries as shown in FIG. 15D. The second prediction is generated by MC using auxiliary sub-blocks covering both the vertical and horizontal boundaries. The size of auxiliary sub-blocks, noted as $w' \times h'$ can be the same as that of the original sub-blocks, i.e., $w' = w$ and $h' = h$. Or the two sizes can be different. In the example of FIG. 15D where $w' = w$ and $h' = h$, the top-left top of an auxiliary sub-block is $(i \times w + w/2, j \times h + h/2)$ and the MV for this auxiliary sub-block is calculated from eq.(1) with $(x,y) = (i \times w + w, j \times h + h)$.

[00125] (iv) For the outer boundaries, motion information used for generating the second prediction can be derived in the same way as the inner boundaries (may be derived at a smaller block size) as shown in FIGS. 15A-15D. Alternatively, the second motion information derived for the nearest inner horizontal/vertical boundary can be used for the horizontal/vertical outer boundary.

[00126] (b) In one example, how to generate the second prediction depends on the width and/or height of the sub-block, and/or block shape, and/or block size, and/or coded mode information.

[00127] (c) In one example, how to generate the second prediction depends on color component.

[00128] (d) In one example, how to generate the second prediction depends on the location of the boundary.

[00129] (e) In one example, the first prediction and the second prediction are from one same reference picture. Alternatively, the second prediction and the first prediction are from different reference pictures.

[00130] Example 4. In one example, the final prediction for a sample along the boundary is calculated as a function of the first prediction and the second prediction signals.

[00131] (a) In one example, the function is defined as linear or non-linear weighted sum.

[00132] (i) In one example, $Pf = (w1 \times P1 + w2 \times P2 + off) / (w1 + w2)$, where Pf is the final prediction; $P1$ and $P2$ are the first and second prediction, respectively; $w1$ and $w2$ are two weighting values; off is the rounding offset, a.e. $off = (w1 + w2) / 2$.

[00133] (ii) In one example, $P_f = (w_1 \times P_1 + w_2 \times P_2 + \text{off}) \gg B$, where P_f is the final prediction; P_1 and P_2 are the first and second prediction, respectively; w_1 and w_2 are two weighting values and $w_1 + w_2 = 2^B$; off is the rounding offset, e.g., $\text{off} = (w_1 + w_2)/2$. Some examples of weighting values (w_1, w_2): (1, 1), (1, 3), (3, 1), (1, 7), (7, 1), (3, 5), (5, 3) and so on.

[00134] (b) In one example, the function is defined as exponential function.

[00135] (c) Weighting values or other function parameters may depend on one or some of the following conditions:

[00136] (i) W and H;

[00137] (ii) w and h;

[00138] (iii) w' and h' ;

[00139] (iv) The position of the sample along the boundary;

[00140] (v) The position of the boundary;

[00141] (vi) The color component;

[00142] (vii) Coding information, including MV, QP, intra-prediction mode, inter-prediction direction, merge mode or AMVP mode, etc.;

[00143] (viii) Distance to the boundary; and/or

[00144] (ix) Continuities at boundary, e.g., gradient values

[00145] Example 5. In one example, besides the second prediction, there may be the third, fourth, or K-th prediction generated for the boundary enhancement.

[00146] (a) In one example, two or more boundary enhancement operations can be conducted in a cascade way. The final prediction output by a previous enhancement operation is input into the following enhancement operation as the first prediction.

[00147] (i) In one example, vertical boundaries are first enhanced as shown in FIG. 15B, with the final prediction P_{f0} as the output. Then horizontal boundaries are enhanced as shown in FIG. 15C with P_{f0} as the first prediction, and the final prediction P_{f1} as the output. P_{f1} is treated as the true final prediction to derive the residues.

[00148] (ii) In one example, horizontal boundaries are first enhanced as shown in FIG. 15C, with the final prediction P_{f0} as the output. Then vertical boundaries are enhanced as shown in FIG. 15B with P_{f0} as the first prediction, and the final prediction P_{f1} as the output. P_{f1} is treated as the true final prediction to derive the residues.

[00149] (iii) Alternatively, furthermore, the order of enhancing vertical (or horizontal) boundaries of different sub-blocks may be defined as raster scan order, or water wave shape, or others.

[00150] (b) In one example, two or more boundary enhancement operations can be conducted in an independent way. The first prediction is noted as $P(0)$, the second to the K th prediction are noted as $P(1) \sim P(K-1)$. $P(0) \sim P(K-1)$ are used jointly to derive the final prediction for a sample along the boundary.

[00151] (i) In one example, $P_f = ((\sum_{r=0}^{K-1} P(r) \times w(r)) + \text{off}) / \sum_{r=0}^{K-1} w(r)$, where $w(r)$ is a weighting value and off is a rounding offset, e.g., $\text{off} = \sum_{r=0}^{K-1} w(r) / 2$

[00152] (ii) In one example, $P_f = ((\sum_{r=0}^{K-1} P(r) \times w(r)) + \text{off}) \gg B$, where $w(r)$ is a weighting value, $\sum_{r=0}^{K-1} w(r) = B$ and off is a rounding offset, e.g., $\text{off} = \sum_{r=0}^{K-1} w(r) / 2$

[00153] Example 6. In one example, whether to apply the boundary enhancement approach and how to generate the second prediction are signaled from the encoder to the decoder. For example, the selection can be signaled in Video Parameter Set (VPS), Sequence Parameter Set (SPS), Picture Parameter Set (PPS), Slice header, Coding Tree Unit (CTU), Coding Tree Block (CTB), Coding Unit (CU) or Prediction Unit (PU), region covering multiple CTU/CTB/CU/PUs.

[00154] Example 7. In one example, embodiments of the disclosed technology may be implemented in conjunction with methods for interweaved prediction. For example, boundary enhancement may be performed for each sub-block based prediction with a specific dividing pattern. Furthermore, the boundaries may be different for different dividing patterns.

[00155] Example 8. The proposed methods may be applied to certain color component.

[00156] (a) In one example, only luma blocks may enable the proposed methods.

[00157] (b) The proposed methods may be applied to certain block sizes/shapes, and/or certain sub-block sizes.

[00158] (c) The proposed methods may be applied to certain coding tools, such as ATMVP and/or affine.

[00159] The examples described above may be incorporated in the context of the methods described below, e.g., method 1600, which may be implemented at a video decoder and/or video encoder.

[00160] FIG. 16 shows a flowchart of an exemplary method for video coding. The method

1600 includes, at step 1610, partitioning a video block into multiple sub-blocks.

[00161] The method 1600 includes, at step 1620, forming a first prediction candidate based on a sub-block based prediction of a plurality of samples in the video block

[00162] The method 1600 includes, at step 1630, forming a second prediction candidate based on an inter prediction of the plurality of samples in a sub-block boundary region. In some embodiments, the first and the second prediction candidates use an identical prediction model, with the first prediction candidate being based on a first subset of the plurality of samples, and the second prediction candidate being based on a second subset of the plurality of samples that is different from the first subset.

[00163] In some embodiments, the sub-block boundary region is an inner boundary of the block of video data, and where the sub-block boundary region includes prediction samples from a neighboring sub-block of the multiple sub-blocks, as described in the context of FIG. 13A. In other embodiments, the sub-block boundary region is an outer boundary of the block of video data, and where the boundary samples include reconstructed samples from a neighboring block of video data, as described in the context of FIG. 13B.

[00164] In some embodiments, and as described in the context of Example 2, the plurality of samples in a sub-block boundary region includes N columns of samples along a vertical boundary and M rows of samples along a horizontal boundary. For example, M or N may be based on dimensions of a sub-block of the multiple sub-blocks. For example, M or N may be based on a type of a component of a sub-block of the multiple sub-blocks (e.g., a luma component or a chroma component). For example, M or N may be based on a location of the sub-block boundary region relative to the video block. For example, M or N may be based on a location of a sub-block of the multiple sub-blocks. In some embodiments, M or N is signaled in a Video Parameter Set (VPS), a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a slice header, a coding tree unit (CTU) or a coding unit (CU).

[00165] In some embodiments, and as described in the context of Example 3, the second prediction candidate may be based on dimensions of a sub-block of the multiple sub-blocks or dimensions of the video block, or a chroma component of a sub-block of the multiple sub-blocks.

[00166] In some embodiments, the sub-block based prediction is identical to the inter prediction. In other embodiments, the first and the second prediction candidates are based on an

identical reference picture. In yet other embodiments, the first prediction candidate is based on a first reference picture, and wherein the second prediction candidate is based on a second reference picture different from the first reference picture.

[00167] The method 1600 includes, at step 1640, forming a final prediction candidate as a function of the first prediction candidate and the second prediction candidate for the plurality of samples in the sub-block boundary region. In some embodiments, and as described in the context of Example 4, the final prediction candidate (Pf) may be a weighted sum of the first prediction candidate (P1) and the second prediction candidate (P2). In one example, $Pf = (w1 \times P1 + w2 \times P2 + offset) / (w1 + w2)$, and in another example, $Pf = (w1 \times P1 + w2 \times P2 + offset) >> B$, where $w1$ and $w2$ are weighting values, where $offset = (w1 + w2) / 2$ is a rounding offset, and where $w1 + w2 = 2^B$. In some embodiments, the final prediction candidate is the same as the first prediction candidate for the plurality of samples not in a sub-block boundary region.

[00168] In some embodiments, the weighting values are based on dimensions of a sub-block of the multiple sub-blocks, dimensions of the video block, a chroma component of the sub-block, one or more properties of the video block, or a location of the sub-block boundary region. For example, the one or more properties include a motion vector, a quantization parameter (QP), an intra-prediction mode, an inter-prediction direction, a merge mode or an advanced motion vector prediction (AMVP) mode.

[00169] The method 1600 includes, at step 1650, processing the video block using the final prediction candidate.

[00170] The method 1600, and as described in the context of Example 5, may further include forming the final prediction candidate further based on one or more additional prediction candidates, each of which is based on a prediction of the plurality of samples in the sub-block boundary region. In some embodiments, the type of the component prediction candidates may be signaled in a Video Parameter Set (VPS), a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a slice header, a coding tree unit (CTU) or a coding unit (CU).

4. Example implementations of the disclosed technology

[00171] FIG. 17 is a block diagram illustrating an example of the architecture for a computer system or other control device 1700 that can be utilized to implement various portions of the presently disclosed technology, including (but not limited to) method 1600. In FIG. 17, the

computer system 1700 includes one or more processors 1705 and memory 1710 connected via an interconnect 1725. The interconnect 1725 may represent any one or more separate physical buses, point to point connections, or both, connected by appropriate bridges, adapters, or controllers. The interconnect 1725, therefore, may include, for example, a system bus, a Peripheral Component Interconnect (PCI) bus, a HyperTransport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), IIC (I2C) bus, or an Institute of Electrical and Electronics Engineers (IEEE) standard 674 bus, sometimes referred to as “Firewire.”

[00172] The processor(s) 1705 may include central processing units (CPUs) to control the overall operation of, for example, the host computer. In certain embodiments, the processor(s) 1705 accomplish this by executing software or firmware stored in memory 1710. The processor(s) 1705 may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic devices (PLDs), or the like, or a combination of such devices.

[00173] The memory 1710 can be or include the main memory of the computer system. The memory 1710 represents any suitable form of random access memory (RAM), read-only memory (ROM), flash memory, or the like, or a combination of such devices. In use, the memory 1710 may contain, among other things, a set of machine instructions which, when executed by processor 1705, causes the processor 1705 to perform operations to implement embodiments of the presently disclosed technology.

[00174] Also connected to the processor(s) 1705 through the interconnect 1725 is a (optional) network adapter 1715. The network adapter 1715 provides the computer system 1700 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.

[00175] FIG. 18 shows a block diagram of an example embodiment of a mobile device 1800 that can be utilized to implement various portions of the presently disclosed technology, including (but not limited to) method 1600. The mobile device 1800 can be a laptop, a smartphone, a tablet, a camcorder, or other types of devices that are capable of processing videos. The mobile device 1800 includes a processor or controller 1801 to process data, and

memory 1802 in communication with the processor 1801 to store and/or buffer data. For example, the processor 1801 can include a central processing unit (CPU) or a microcontroller unit (MCU). In some implementations, the processor 1801 can include a field-programmable gate-array (FPGA). In some implementations, the mobile device 1800 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 1802 can include and store processor-executable code, which when executed by the processor 1801, configures the mobile device 1800 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.

[00176] To support various functions of the mobile device 1800, the memory 1802 can store information and data, such as instructions, software, values, images, and other data processed or referenced by the processor 1801. 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 1802. In some implementations, the mobile device 1800 includes an input/output (I/O) unit 1803 to interface the processor 1801 and/or memory 1802 to other modules, units or devices. For example, the I/O unit 1803 can interface the processor 1801 and memory 1802 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 1800 can interface with other devices using a wired connection via the I/O unit 1803. The mobile device 1800 can also interface with other external interfaces, such as data storage, and/or visual or audio display devices 1804, 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 1804 or an external device. For example, the display device 1804 can display a video frame that includes a block (a CU, PU or TU) that applies the intra-block copy based on whether the block is encoded using a motion compensation algorithm, and in accordance with the disclosed technology.

[00177] In some embodiments, a video decoder apparatus may implement a method of sub-

block based prediction as described herein is used for video decoding. The various features of the method may be similar to the above-described method 1600.

[00178] 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. 17 and FIG. 18.

[00179] Various embodiments and techniques disclosed in the present document can be described in the following listing of examples.

[00180] 1. A video processing method (e.g., method 1900 depicted in FIG. 19), comprising: partitioning (1902) a video block into multiple sub-blocks; forming (1904) a final prediction candidate as a function of a first prediction candidate corresponding to a sub-block based prediction of a plurality of samples in a sub-block boundary region and a second prediction candidate corresponding to an inter prediction of the plurality of samples in the sub-block boundary region; and processing (1906) the video block using the final prediction candidate. For example, the processing may include, at a video encoder, generating a coded representation of the video block in which the video block is coded using a predictive coding of the video block. For example, the processing may include, at a video decoder, the final prediction candidate to perform motion compensation to generate the video block.

[00181] 2. The method of example 1, wherein the sub-block boundary region comprises an inner boundary of the video block.

[00182] 3. The method of example 1, wherein the sub-block boundary region is an outer boundary of the video block.

[00183] 4. The method of example 1, wherein the plurality of samples in the sub-block boundary region comprises N columns of samples along a vertical boundary and M rows of samples along a horizontal boundary.

[00184] 5. The method of example 4, wherein M or N is based on dimensions of a sub-block of the multiple sub-blocks.

[00185] 6. The method of example 4, wherein M or N is based on a type of a colour component of a sub-block of the multiple sub-blocks.

[00186] 7. The method of example 4, wherein M or N is based on a location of the sub-block boundary region relative to the video block.

[00187] 8. The method of example 4, wherein M or N is based on a location of a sub-block of the multiple sub-blocks.

[00188] 9. The method of example 4, wherein M or N is signaled in a Video Parameter Set (VPS), a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a slice header, a coding tree unit (CTU) or a coding unit (CU).

[00189] 10. The method of example 1, wherein the first and the second prediction candidates use an identical prediction model, wherein the first prediction candidate is based on a first subset of the plurality of samples, and wherein the second prediction candidate is based on a second subset of the plurality of samples that is different from the first subset.

[00190] 11. The method of example 1 or 10, wherein the second prediction candidate is based on dimensions of a sub-block of the multiple sub-blocks or dimensions of the video block.

[00191] 12. The method of example 1 or 10, wherein the second prediction candidate is based on a colour component of a sub-block of the multiple sub-blocks.

[00192] 13. The method of example 1, wherein the sub-block based prediction is identical to the inter prediction.

[00193] 14. The method of example 1, wherein the first and the second prediction candidates are based on an identical reference picture.

[00194] 15. The method of example 1, wherein the first prediction candidate is based on a first reference picture, and wherein the second prediction candidate is based on a second reference picture different from the first reference picture.

[00195] 16. The method of example 1, wherein the final prediction candidate (Pf) is a weighted sum of the first prediction candidate (P1) and the second prediction candidate (P2).

[00196] 17. The method of example 16, wherein $Pf = (w1 \times P1 + w2 \times P2 + \text{offset}) / (w1 + w2)$, wherein w1 and w2 are weighting values, and wherein $\text{offset} = (w1 + w2) / 2$ is a rounding offset.

[00197] 18. The method of example 17, wherein the weighting values are based on dimensions of a sub-block of the multiple sub-blocks, dimensions of the video block, a chroma component of the sub-block, one or more properties of the video block, or a location of the sub-block boundary region.

[00198] 19. The method of example 18, wherein the one or more properties comprise a motion vector, a quantization parameter (QP), an intra-prediction mode, an inter-prediction direction, a

merge mode or an advanced motion vector prediction (AMVP) mode.

[00199] 20. The method of example 16, wherein $Pf = (w1 \times P1 + w2 \times P2 + \text{offset}) \gg B$, wherein $w1$ and $w2$ are weighting values, wherein $\text{offset} = (w1 + w2)/2$ is a rounding offset, and wherein $w1 + w2 = 2B$.

[00200] 21. The method of example 1, further comprising: forming one or more additional prediction candidates based on a prediction of the plurality of samples in the sub-block boundary region, wherein the final prediction candidate is further based on the one or more additional prediction candidates.

[00201] 22. The method of example 1, wherein a type of the second prediction candidate is signaled in a Video Parameter Set (VPS), a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a slice header, a coding tree unit (CTU) or a coding unit (CU).

[00202] 23. The method of example 1, wherein a chroma component of a sub-block comprises the plurality of samples in the sub-block boundary region.

[00203] 24. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is affine prediction.

[00204] 25. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is alternative temporal motion vector prediction (ATMVP).

[00205] 26. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is spatial-temporal motion vector prediction (STMVP).

[00206] 27. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is bi-directional optical flow (BIO).

[00207] 28. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Frame-Rate Up Conversion (FRUC).

[00208] 29. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Locally Adaptive Motion Vector Resolution (LAMVR).

[00209] 30. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Temporal Motion Vector Prediction (TMVP).

[00210] 31. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Overlapped Block Motion Compensation (OBMC),

[00211] 32. The method of example 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Decoder-side Motion Vector Refinement (DMVR).

[00212] 33. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of examples 1 to 32.

[00213] 34. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of examples 1 to 32.

[00214] 35. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method recited in one or more of examples 1 to 32.

[00215] From the foregoing, it will be appreciated that specific embodiments of the presently disclosed technology have been described herein for purposes of illustration, but that various modifications may be made without deviating from the scope of the invention. Accordingly, the presently disclosed technology is not limited except as by the appended claims.

[00216] Implementations of the subject matter and the functional operations described in this patent document can be implemented in various systems, digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a tangible and non-transitory computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more of them. The term “data processing unit” or “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can

include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[00217] 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.

[00218] The processes and logic flows described in this specification 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).

[00219] 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 nonvolatile memory, media and memory devices, including by way of

example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[00220] It is intended that the specification, together with the drawings, be considered exemplary only, where exemplary means an example. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. Additionally, the use of “or” is intended to include “and/or”, unless the context clearly indicates otherwise.

[00221] 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.

[00222] 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.

[00223] 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 video processing method, comprising:
partitioning a video block into multiple sub-blocks;
forming a final prediction candidate as a function of a first prediction candidate corresponding to a sub-block based prediction of a plurality of samples in a sub-block boundary region and a second prediction candidate corresponding to an inter prediction of the plurality of samples in the sub-block boundary region; and
processing the video block using the final prediction candidate.
2. The method of claim 1, wherein the sub-block boundary region comprises an inner boundary of the video block.
3. The method of claim 1, wherein the sub-block boundary region is an outer boundary of the video block.
4. The method of claim 1, wherein the plurality of samples in the sub-block boundary region comprises N columns of samples along a vertical boundary and M rows of samples along a horizontal boundary.
5. The method of claim 4, wherein M or N is based on dimensions of a sub-block of the multiple sub-blocks.
6. The method of claim 4, wherein M or N is based on a type of a colour component of a sub-block of the multiple sub-blocks.
7. The method of claim 4, wherein M or N is based on a location of the sub-block boundary region relative to the video block.
8. The method of claim 4, wherein M or N is based on a location of a sub-block of the multiple sub-blocks.

9. The method of claim 4, wherein M or N is signaled in a Video Parameter Set (VPS), a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a slice header, a coding tree unit (CTU) or a coding unit (CU).
10. The method of claim 1, wherein the first and the second prediction candidates use an identical prediction model, wherein the first prediction candidate is based on a first subset of the plurality of samples, and wherein the second prediction candidate is based on a second subset of the plurality of samples that is different from the first subset.
11. The method of claim 1 or 10, wherein the second prediction candidate is based on dimensions of a sub-block of the multiple sub-blocks or dimensions of the video block.
12. The method of claim 1 or 10, wherein the second prediction candidate is based on a colour component of a sub-block of the multiple sub-blocks.
13. The method of claim 1, wherein the sub-block based prediction is identical to the inter prediction.
14. The method of claim 1, wherein the first and the second prediction candidates are based on an identical reference picture.
15. The method of claim 1, wherein the first prediction candidate is based on a first reference picture, and wherein the second prediction candidate is based on a second reference picture different from the first reference picture.
16. The method of claim 1, wherein the final prediction candidate (Pf) is a weighted sum of the first prediction candidate (P1) and the second prediction candidate (P2).
17. The method of claim 16, wherein $Pf = (w1 \times P1 + w2 \times P2 + offset) / (w1 + w2)$, wherein $w1$ and $w2$ are weighting values, and wherein $offset = (w1 + w2) / 2$ is a rounding offset.
18. The method of claim 17, wherein the weighting values are based on dimensions of a sub-block of the multiple sub-blocks, dimensions of the video block, a chroma component of the sub-block, one or more properties of the video block, or a location of the sub-block boundary region.

19. The method of claim 18, wherein the one or more properties comprise a motion vector, a quantization parameter (QP), an intra-prediction mode, an inter-prediction direction, a merge mode or an advanced motion vector prediction (AMVP) mode.
20. The method of claim 16, wherein $Pf = (w1 \times P1 + w2 \times P2 + offset) >> B$, wherein $w1$ and $w2$ are weighting values, wherein $offset = (w1 + w2) / 2$ is a rounding offset, and wherein $w1 + w2 = 2^B$.
21. The method of claim 1, further comprising:
forming one or more additional prediction candidates based on a prediction of the plurality of samples in the sub-block boundary region, wherein the final prediction candidate is further based on the one or more additional prediction candidates.
22. The method of claim 1, wherein a type of the second prediction candidate is signaled in a Video Parameter Set (VPS), a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), a slice header, a coding tree unit (CTU) or a coding unit (CU).
23. The method of claim 1, wherein a chroma component of a sub-block comprises the plurality of samples in the sub-block boundary region.
24. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is affine prediction.
25. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is alternative temporal motion vector prediction (ATMVP).
26. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is spatial-temporal motion vector prediction (STMVP).
27. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is bi-directional optical flow (BIO).
28. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Frame-Rate Up Conversion (FRUC).

29. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Locally Adaptive Motion Vector Resolution (LAMVR).
30. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Temporal Motion Vector Prediction (TMVP).
31. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Overlapped Block Motion Compensation (OBMC).
32. The method of claim 1, wherein a prediction mode for the first prediction candidate and the second prediction candidate is Decoder-side Motion Vector Refinement (DMVR).
33. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of claims 1 to 32.
34. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of claims 1 to 32.
35. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method recited in one or more of claims 1 to 32.

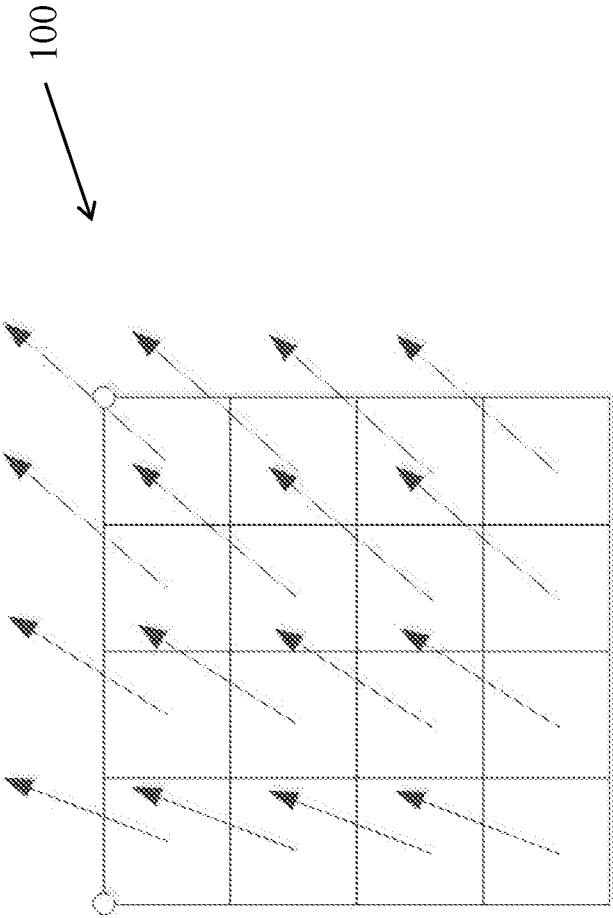


FIG. 1

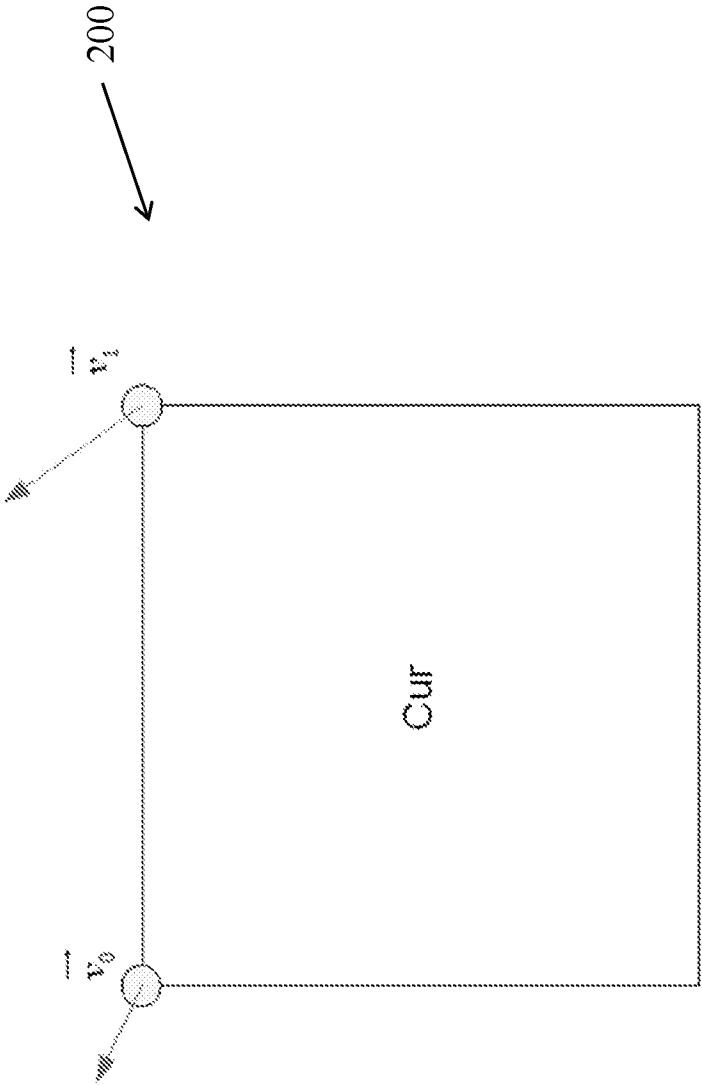


FIG. 2

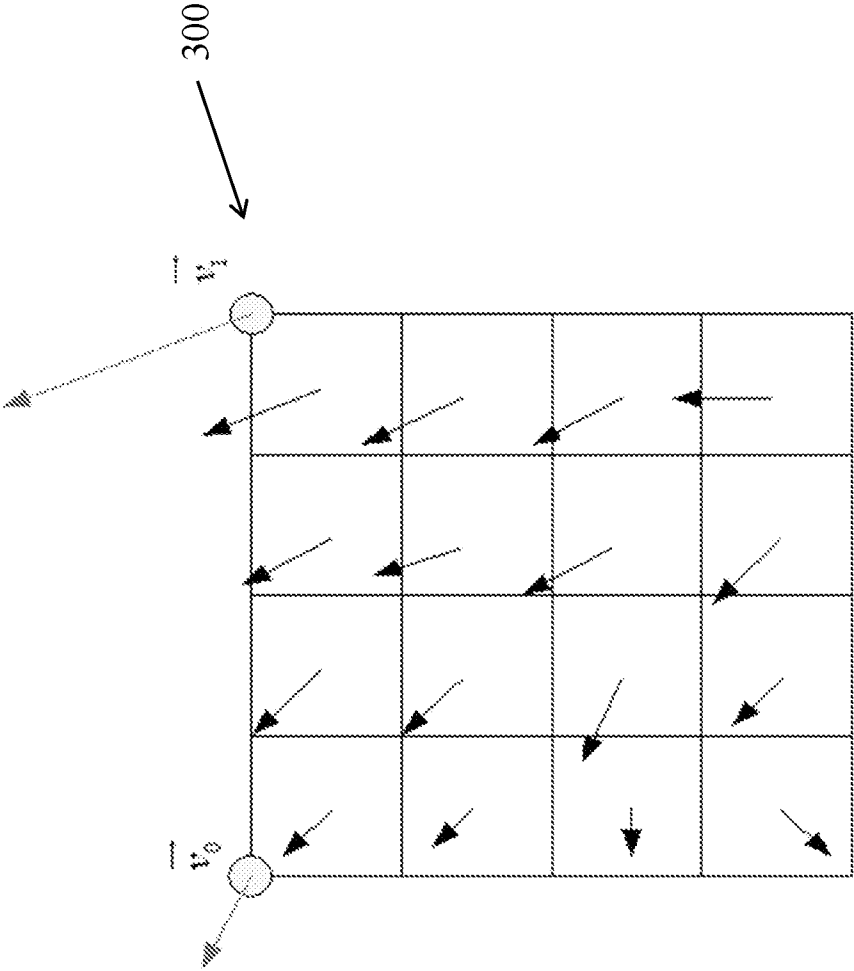


FIG. 3

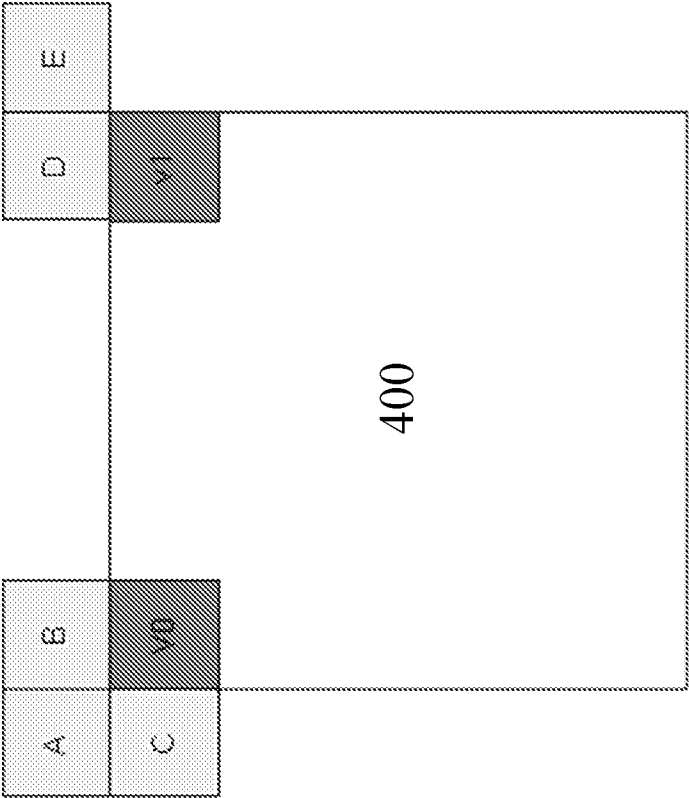


FIG. 4

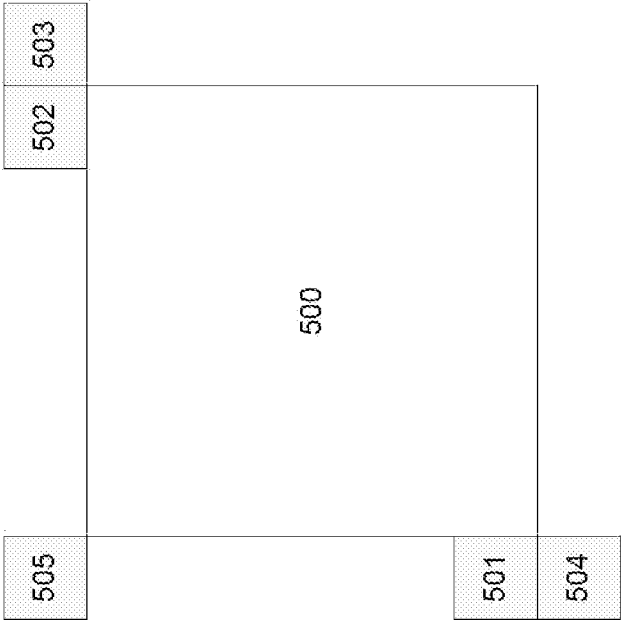


FIG. 5A

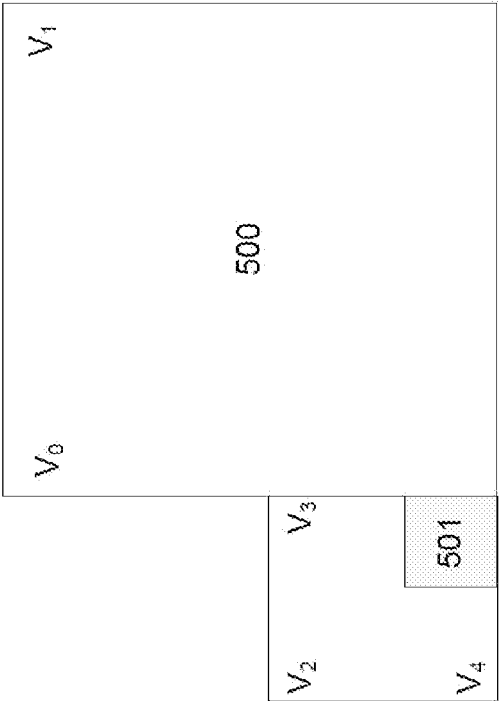


FIG. 5B

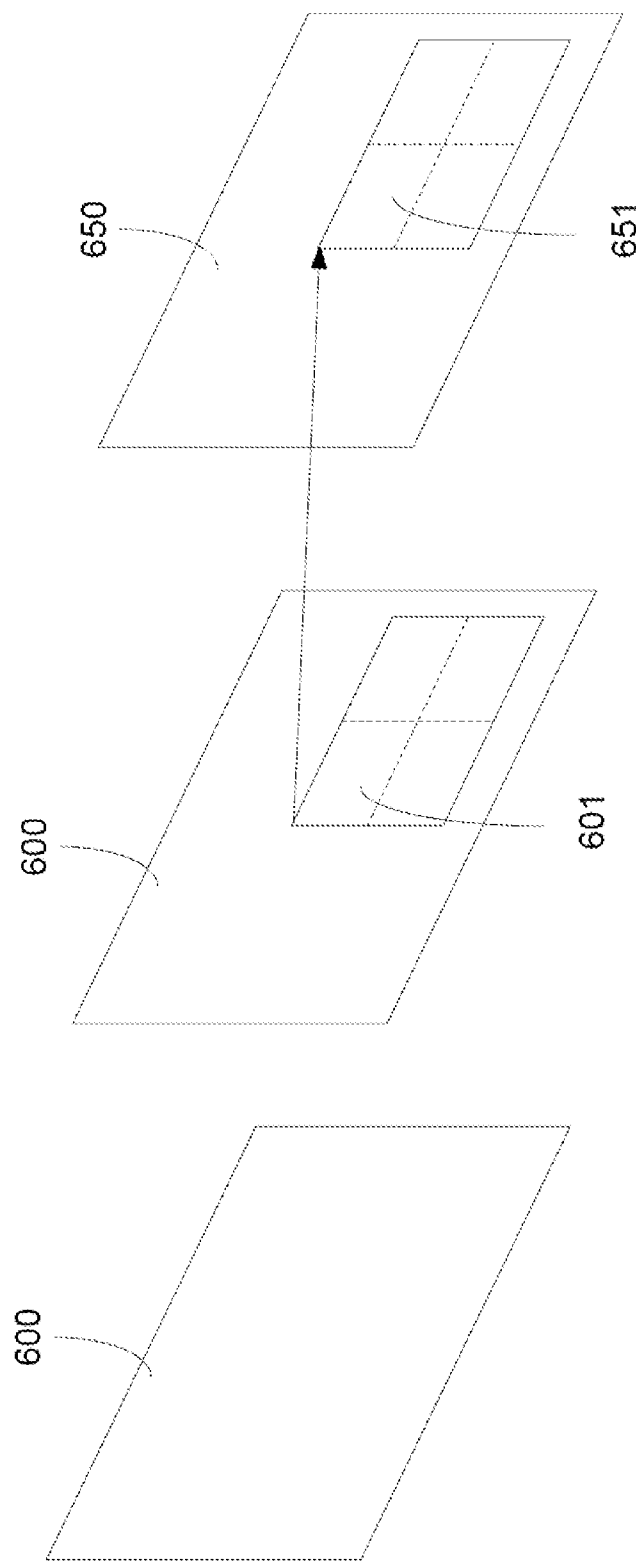


FIG. 6

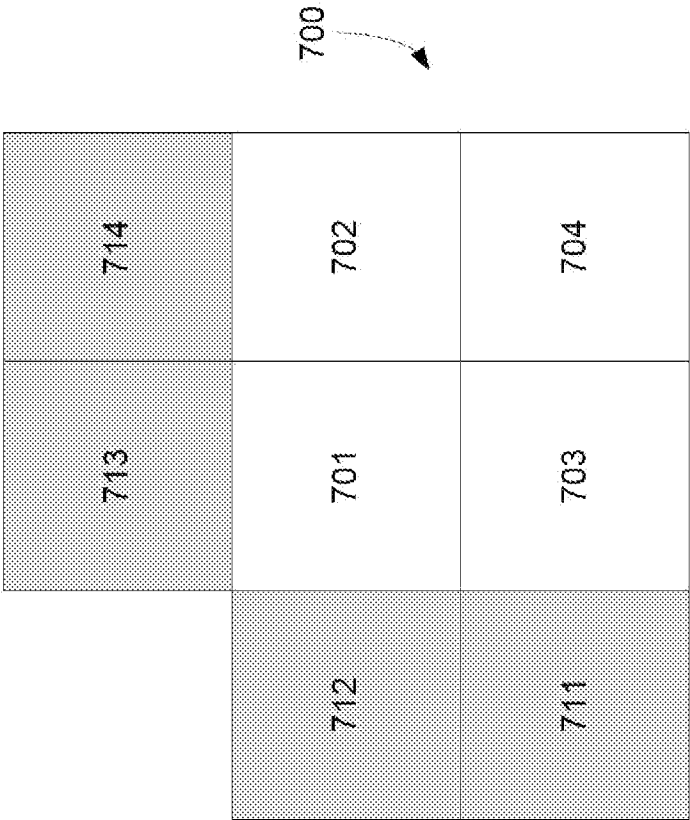


FIG. 7

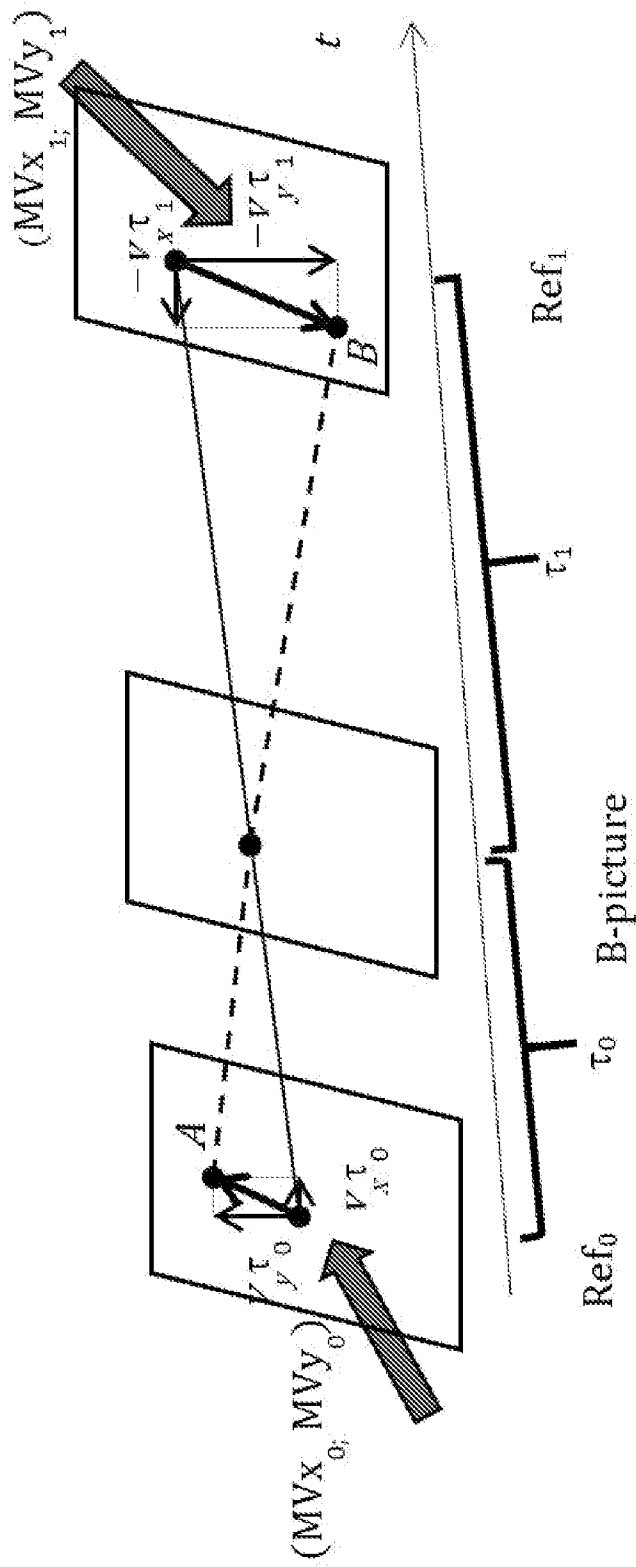


FIG. 8

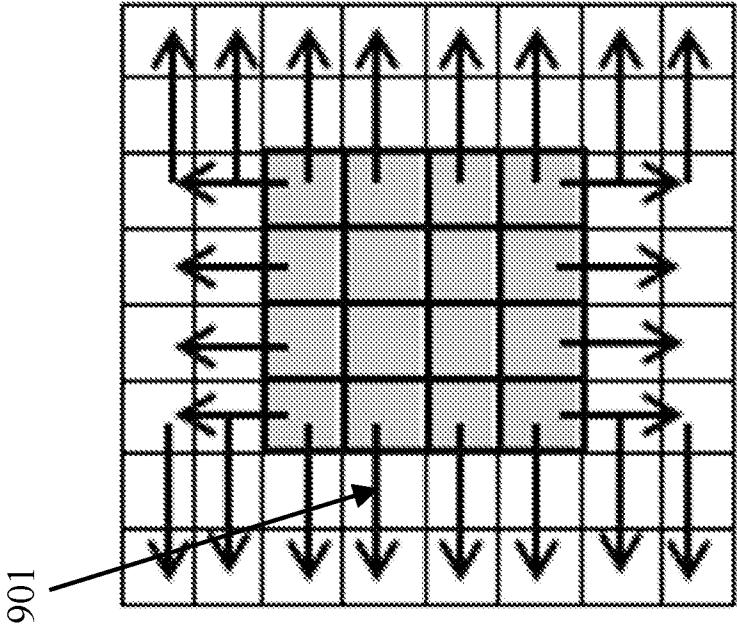


FIG. 9B

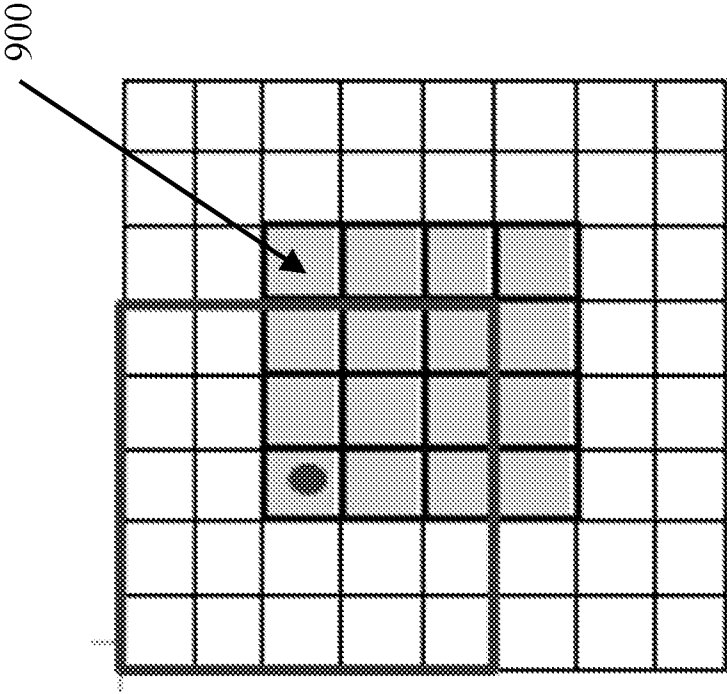


FIG. 9A

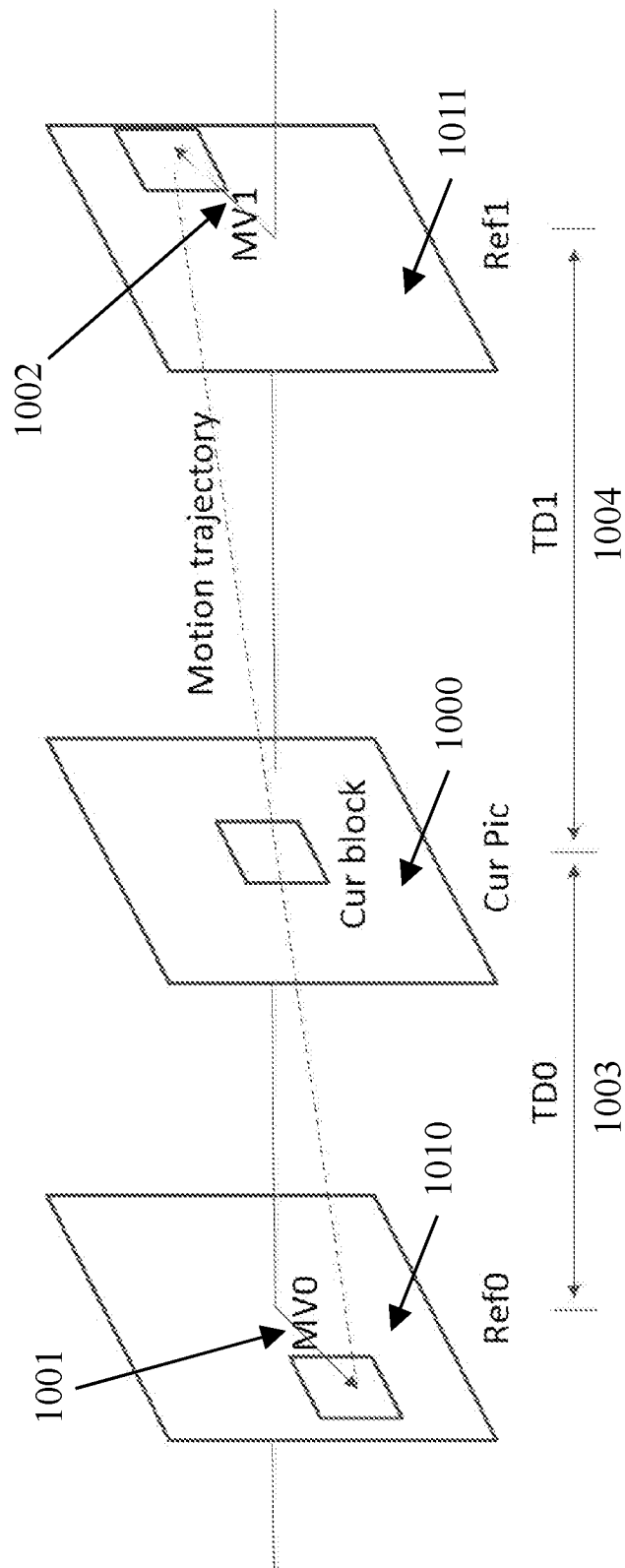


FIG. 10

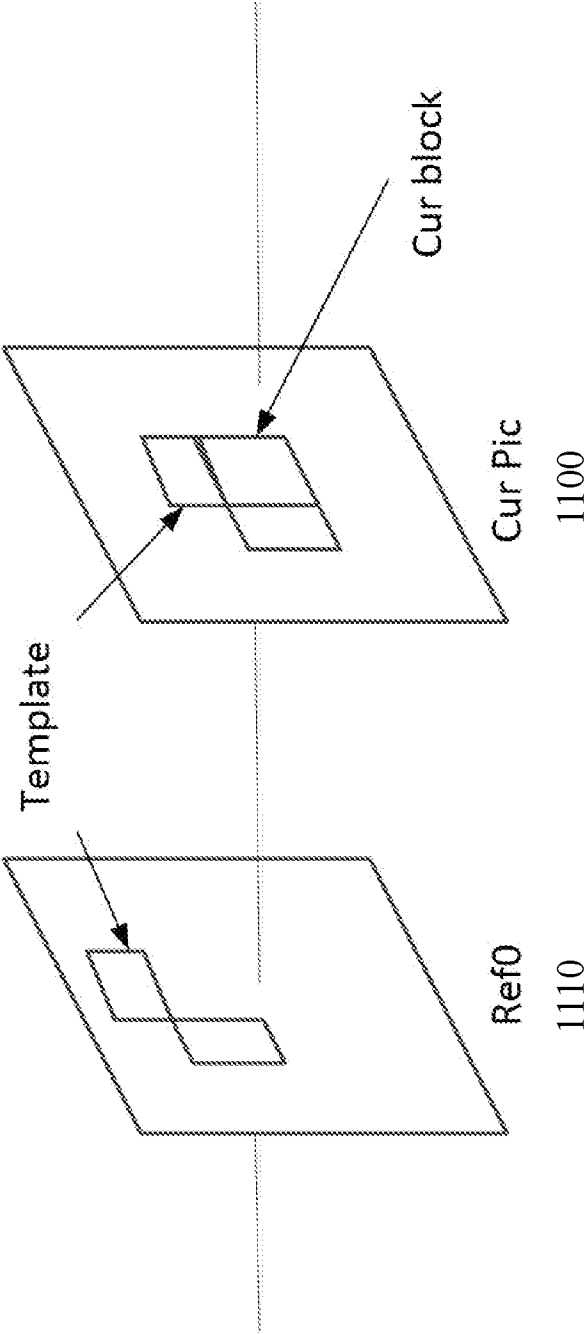


FIG. 11

1200

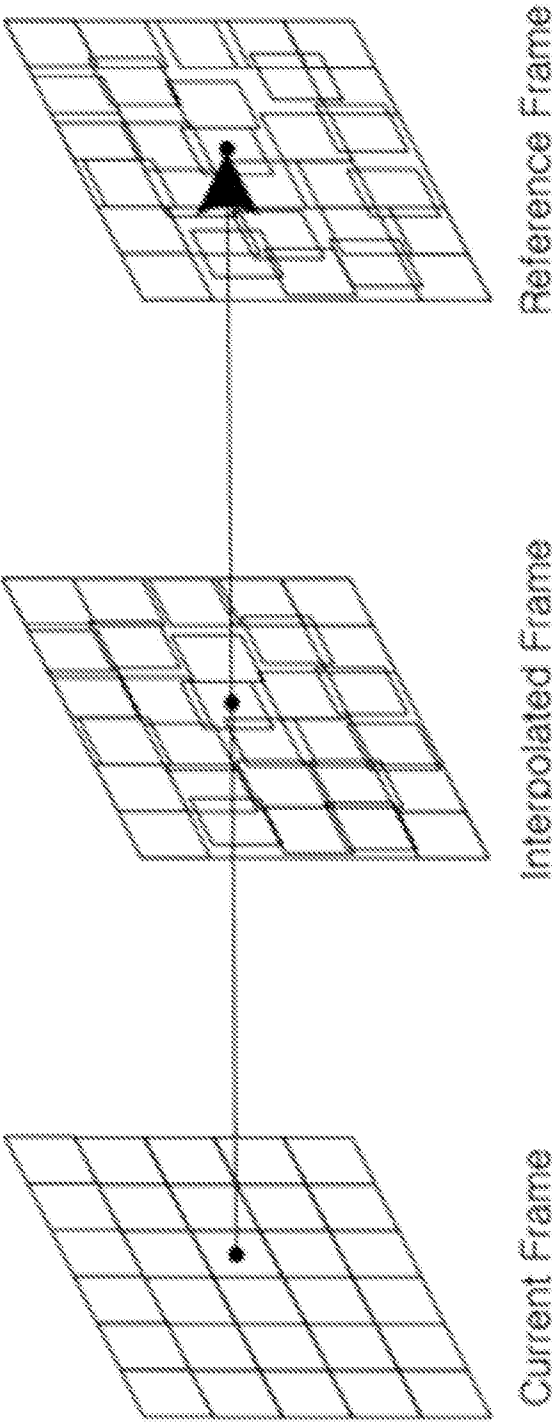


FIG. 12

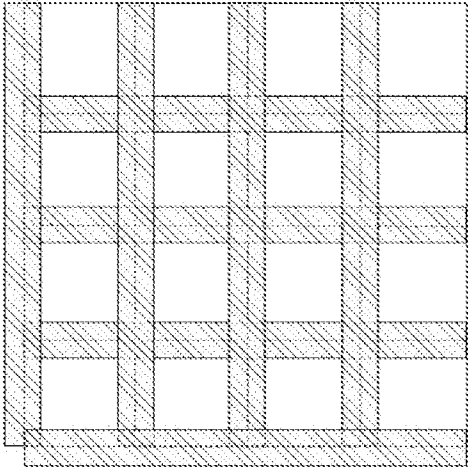


FIG. 13B

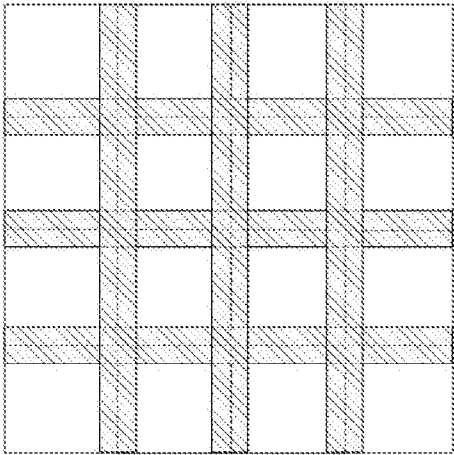


FIG. 13A

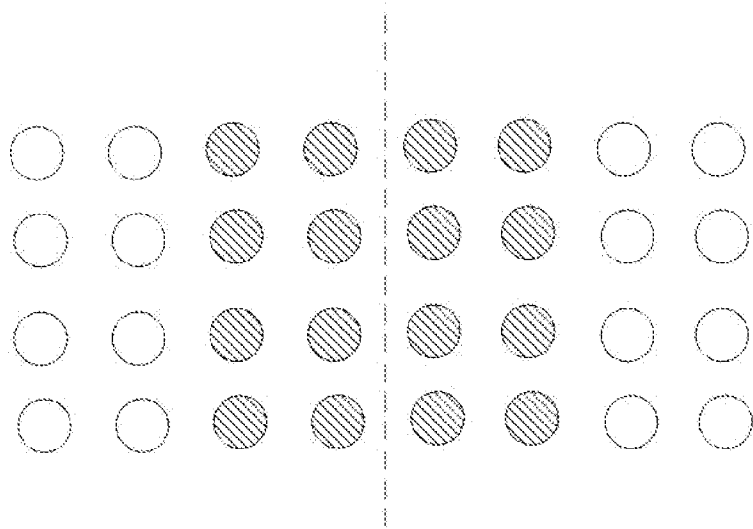


FIG. 14B

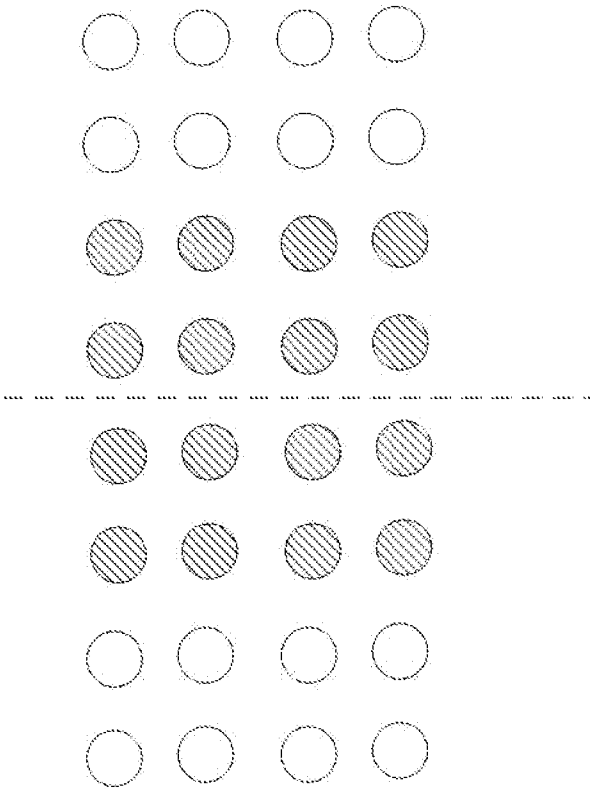


FIG. 14A

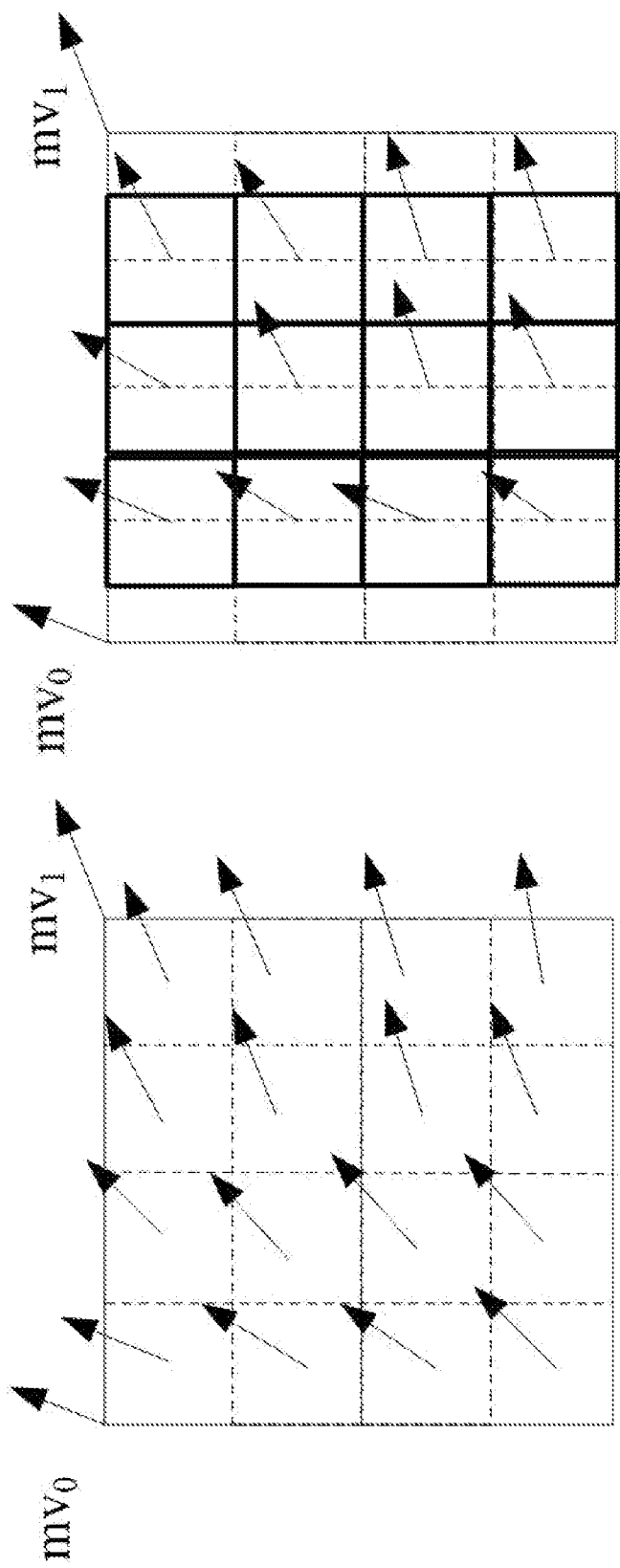


FIG. 15A

FIG. 15B

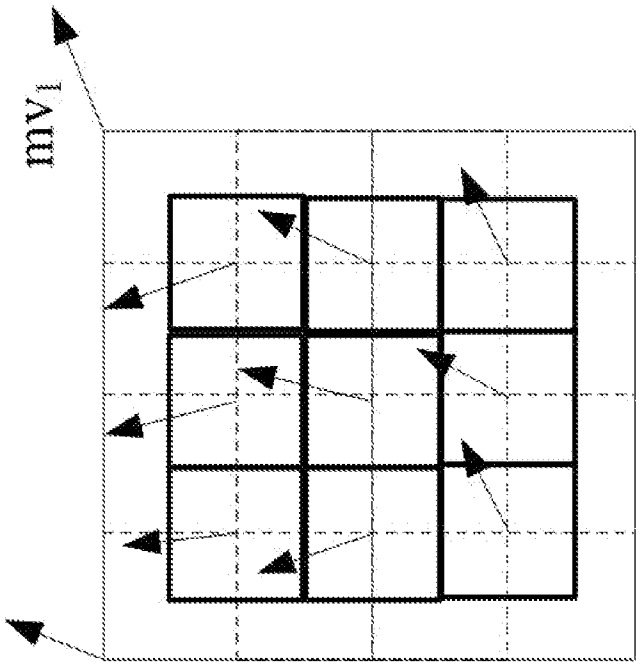


FIG. 15D

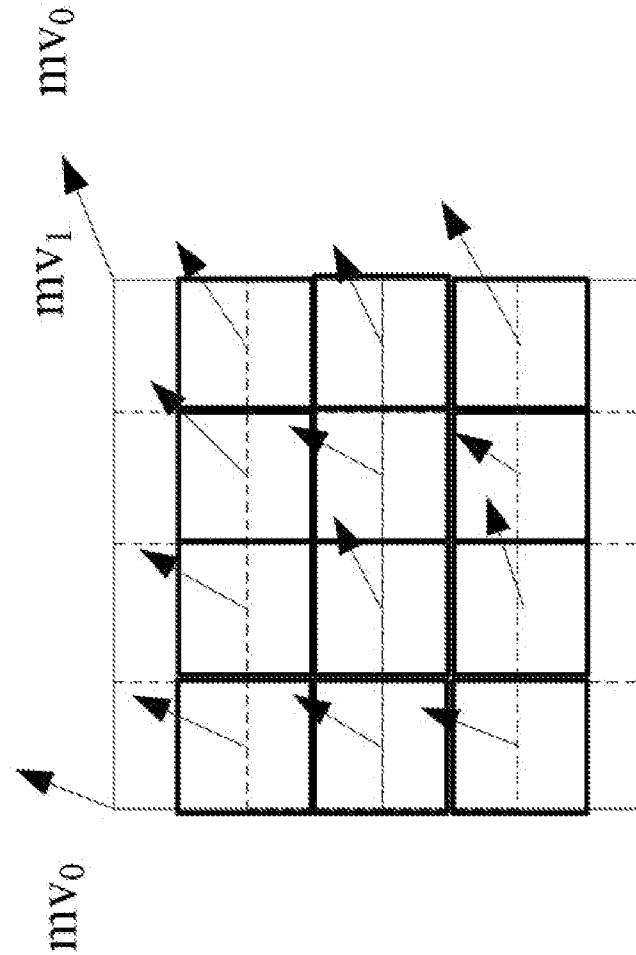


FIG. 15C

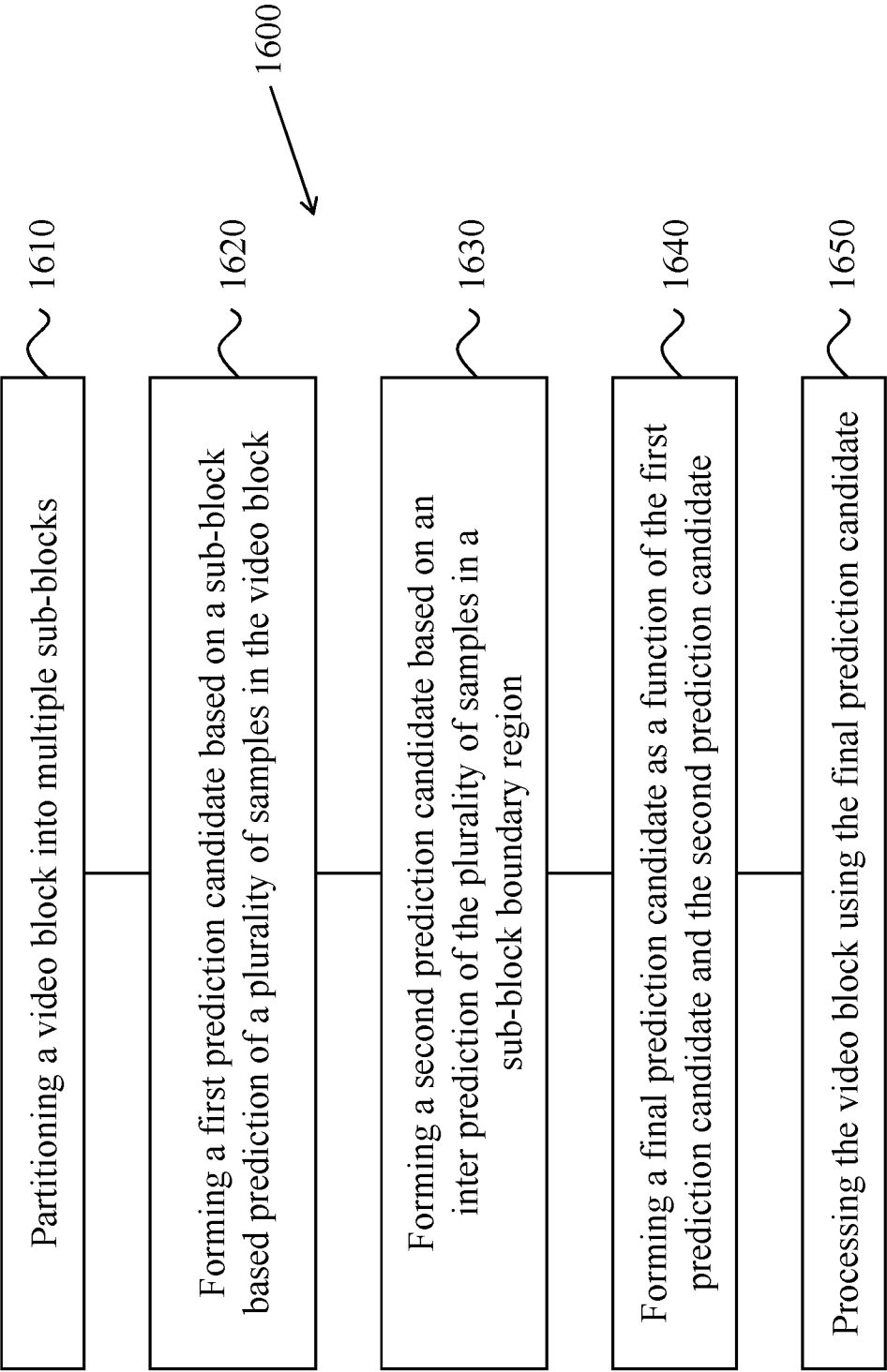


FIG. 16

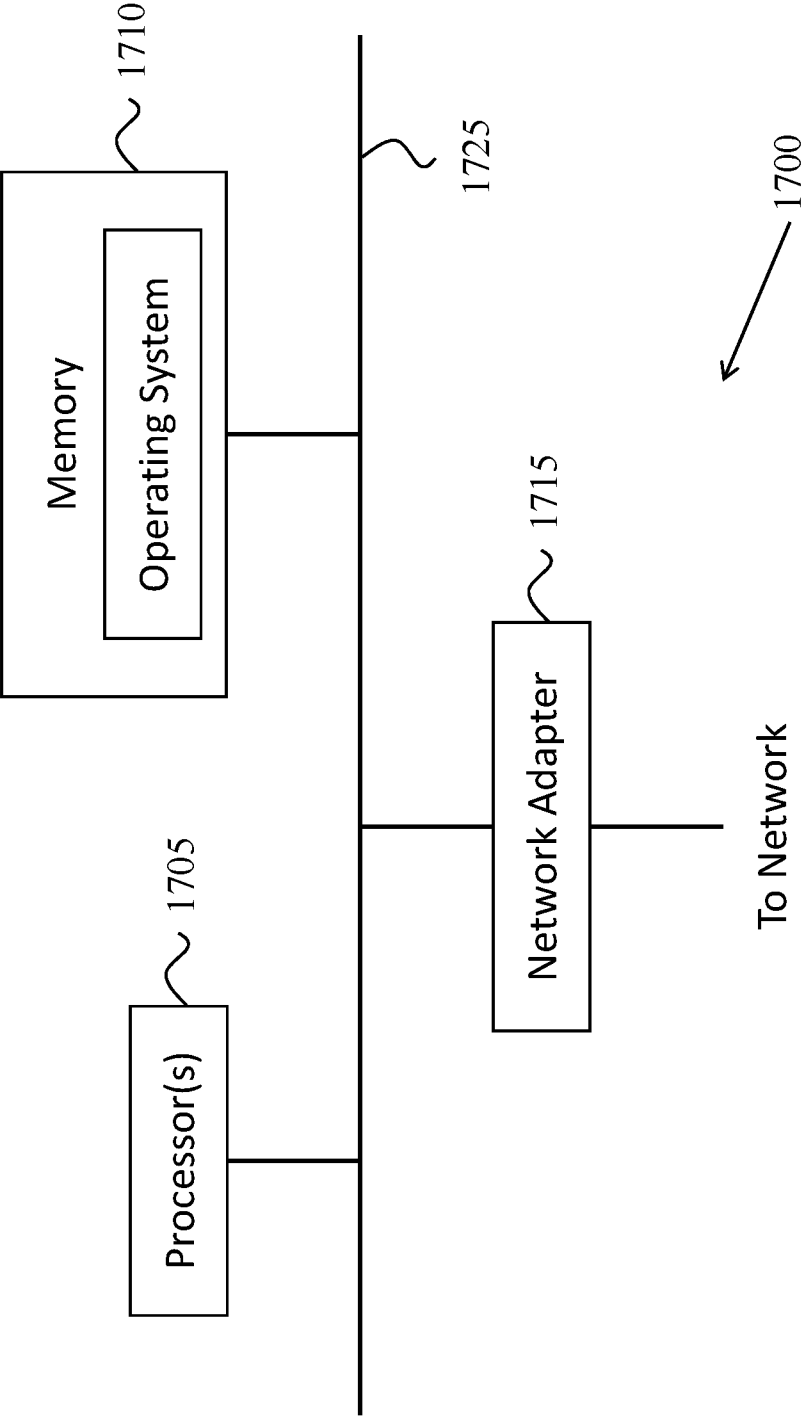


FIG. 17

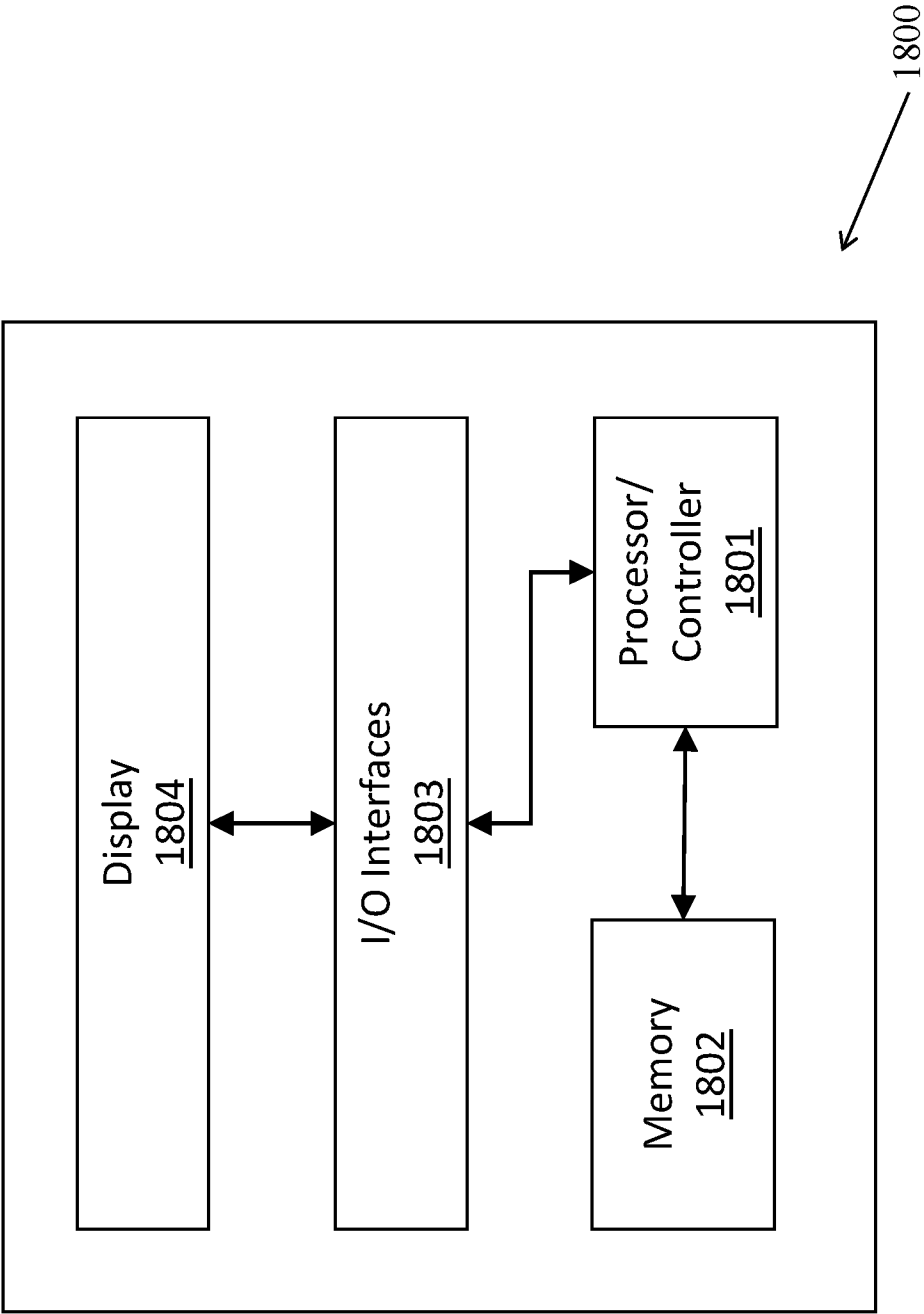


FIG. 18

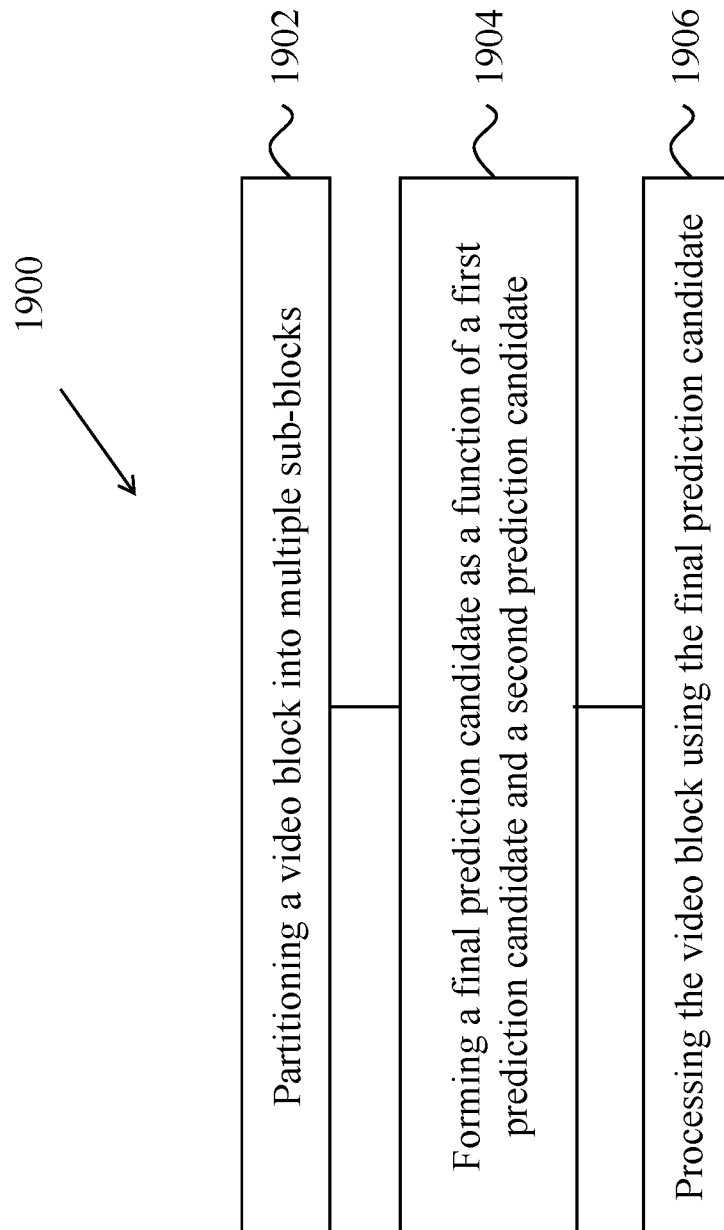


FIG. 19

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2019/055548

A. CLASSIFICATION OF SUBJECT MATTER INV. H04N19/573 H04N19/105 H04N19/139 H04N19/159 H04N19/176 ADD.		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H04N		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, INSPEC, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	BORDES P ET AL: "Description of SDR, HDR and 360° video coding technology proposal by Qualcomm and Technicolor " medium complexity version", 10. JVET MEETING; 10-4-2018 - 20-4-2018; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-J0022-v3, 12 April 2018 (2018-04-12), XP030151186, page 19 - page 29	1,4-6,9, 11-15, 33-35
A	----- -/--	2,3,7,8, 10,16-32
☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 28 August 2019		Date of mailing of the international search report 04/09/2019
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Cyranka, Oliver

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2019/055548

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHEN H ET AL: "Description of SDR, HDR and 360° video coding technology proposal by Huawei, GoPro, HiSilicon, and Samsung " general application scenario", 10. JVET MEETING; 10-4-2018 - 20-4-2018; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-J0025-v4, 14 April 2018 (2018-04-14), XP030151192, page 30 - page 66	1,10, 16-35
A	-----	2-9, 11-15
X	XIU X ET AL: "Description of SDR, HDR and 360° video coding technology proposal by InterDigital Communications and Dolby Laboratories", 10. JVET MEETING; 10-4-2018 - 20-4-2018; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-J0015, 3 April 2018 (2018-04-03), XP030151174, page 3 - page 14	1-3,7,8, 33-35
A	-----	4-6,9-32
X,P	ZHANG (BYTEDANCE) K ET AL: "CE4-related: Interweaved Prediction for Affine Motion Compensation", 11. JVET MEETING; 20180711 - 20180718; LJUBLJANA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-K0102 11 July 2018 (2018-07-11), XP030199404, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0102-v2.zip JVET-K0102.docx [retrieved on 2018-07-11] the whole document ----- -/--	1-35

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2019/055548

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	Y-C LIN ET AL: "CE10.2.4: Subblock OBMC with integer-MV-based OBMC at CU boundaries", 13. JVET MEETING; 20190109 - 20190118; MARRAKECH; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-M0181 9 January 2019 (2019-01-09), XP030201073, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/13_Marrakech/wg11/JVET-M0181-v3.zip JVET-M0181-v1.docx [retrieved on 2019-01-09] the whole document -----	1-35