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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
04 June 2020 (04.06.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/112451 A1

(51) International Patent Classification:
H04N 19/52 (2014.01) *H04N 19/513* (2014.01)

(21) International Application Number:
PCT/US2019/062346

(22) International Filing Date:
20 November 2019 (20.11.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
18306572.1 27 November 2018 (27.11.2018) EP

(71) Applicant: **INTERDIGITAL VC HOLDINGS, INC.** [US/US]; 200 Bellevue Parkway, Suite 300, Wilmington, Delaware 19809 (US).

(72) Inventors: **ROBERT, Antoine**; Technicolor R&D France, 975 Avenue des Champs Blancs, Zac des Champs Blancs - CS 17616, 35576 Cesson-Sevigne (FR). **GALPIN, Franck**; Technicolor R&D France, 975 Avenue des Champs Blancs, Zac des Champs Blancs - CS 17616, 35576 Cesson-Sevigne (FR). **BORDES, Philippe**; Technicolor R&D France, 975 Avenue des Champs Blancs, Zac des Champs Blancs - CS 17616, 35576 Cesson-Sevigne (FR).

(74) Agent: **DORINI, Brian J.** et al.; Interdigital VC Holdings, Inc., Four Research Way, 3rd Floor, Princeton, New Jersey 08540 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, 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).

(54) Title: COMBINING AFFINE CANDIDATES

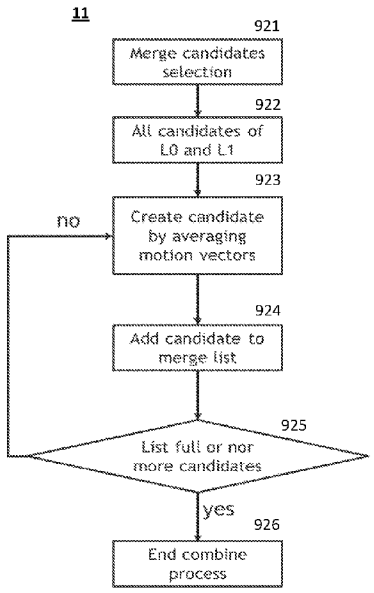

Figure 11

(57) Abstract: At least a method and an apparatus are provided for efficiently encoding or decoding video. For example, a first set of affine motion model candidates is obtained for a current block. A second set of affine motion model candidates is obtained using a combination of one or more of the affine motion model candidates in the first set. The current block is then encoded or decoded based on one or more of the affine motion model candidates in the first set and the second set.

WO 2020/112451 A1

[Continued on next page]

Declarations under Rule 4.17:

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

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

Published:

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

COMBINING AFFINE CANDIDATES

TECHNICAL FIELD

At least one of the present embodiments generally relates to a method or an apparatus for video encoding or decoding, and more particularly, to a method or an apparatus for efficiently providing video compression and/or decompression with additional affine motion model candidates obtained from a combination of other affine motion model candidates.

BACKGROUND

To achieve high compression efficiency, image and video coding schemes usually employ prediction, including motion vector prediction, and transform to leverage spatial and temporal redundancy in the video content. Generally, intra or inter prediction is used to exploit the intra or inter frame correlation, then the differences between the original image and the predicted image, often denoted as prediction errors or prediction residuals, are transformed, quantized, and entropy coded. To reconstruct the video, the compressed data are decoded by inverse processes corresponding to the entropy coding, quantization, transformation, and prediction.

Recent additions to video compression technology include various industry standards, versions of the reference software and/or documentations such as Joint Exploration Model (JEM) and later VTM (Versatile Video Coding (VVC) Test Model) being developed by the JVET (Joint Video Exploration Team) group. The aim is to make further improvements to the existing HEVC (High Efficiency Video Coding) standard.

SUMMARY

The drawbacks and disadvantages of the prior art are solved and addressed by one or more aspects described herein.

According to an embodiment, a method for video encoding is provided, comprising: obtaining a first set of affine motion model candidates for a current block; obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and encoding the current block based on one or more of the affine motion model candidates in the first set and the second set.

According to another embodiment, a method for video decoding is provided, comprising: obtaining a first set of affine motion model candidates for a current block; obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and decoding the current block based on one or more of the affine motion model candidates in the first set and the second set.

According to another embodiment, an apparatus for video encoding is provided, comprising: means for obtaining a first set of affine motion model candidates for a current block; means for obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and means for encoding the current block based on one or more of the affine motion model candidates in the first set and the second set.

According to another embodiment, an apparatus for video decoding is provided, comprising: means for obtaining a first set of affine motion model candidates for a current block; means for obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and means for decoding the current block based on one or more of the affine motion model candidates in the first set and the second set.

According to another embodiment, an apparatus for video encoding is presented, comprising one or more processors, wherein said one or more processors are configured to: obtain a first set of affine motion model candidates for a current block; obtain a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and encode the current block based on based on one or more of the affine motion model candidates in the first set and the second set.

According to another embodiment, an apparatus for video decoding is presented, comprising one or more processors, wherein said one or more processors are configured to: obtain a first set of affine motion model candidates for a current block; obtain a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and decode the current block based on one or more of the affine motion model candidates in the first set and the second set.

According to another embodiment, a signal comprising encoded video is formed by performing: obtaining a first set of affine motion model candidates for a current block; obtaining a

second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; encoding the current block based on one or more of the affine motion model candidates in the first set and the second set; and forming the bitstream comprising the encoded current block.

According to another embodiment, the combination of one or more of the affine motion model candidates in the first set comprises pair-wise averaging of two of the affine motion model candidates in the first set.

According to another embodiment, the first set comprises at least a first list and a second list of affine motion model candidates.

According to another embodiment, each of the first set and the second set of affine motion model candidates comprise two or three motion vectors.

According to another embodiment, the pair-wise averaging is performed using the affine motion model candidates within the first list or within the second list.

According to another embodiment, selection order of the one or more of the affine motion model candidates in the first set to be combined is predetermined.

According to another embodiment, the combination is based on a combination of different motion vector components from different affine motion model candidates in the first set.

According to another embodiment, the combination is based on a combination of the affine motion model candidates in the first set having different reference pictures.

According to another embodiment, the motion vector components represent control point motion vectors.

Additionally, an embodiment provides a computer program product comprising instructions which when executed by one or more processors cause the one or more processors to perform the encoding method or decoding method according to any of the embodiments described above. One or more of the present embodiments also provide a computer readable storage medium having stored thereon instructions for encoding or decoding video data according to the methods described above. One or more embodiments also provide a computer readable storage medium having stored thereon a bitstream generated according to the methods described above. One or more embodiments also provide a method and apparatus for transmitting or receiving the bitstream generated according to the methods described above.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a Coding Tree Unit and Coding Tree concepts to represent a compressed HEVC picture.

Figure 2 illustrates division of a Coding Tree Unit into Coding Units, Prediction Units and Transform Units.

Figure 3 illustrates examples of affine motion models.

Figure 4 illustrates 4x4 sub-CU based affine motion vector fields used in two affine motion models.

Figure 5 illustrates a motion vector prediction process for affine inter CUs.

Figure 6 illustrates an exemplary process for creating new affine motion model candidates in affine AMVP (Advanced Motion Vector Prediction) mode.

Figure 7 illustrates motion vector prediction candidates in affine merge.

Figure 8 illustrates spatial derivation of affine motion field control points in affine merge.

Figure 9 illustrates an exemplary process to create new affine candidates in affine merge mode.

Figure 10 illustrates an exemplary process to combine candidates.

Figure 11 illustrates an exemplary pair-wise process.

Figure 12 illustrates an exemplary process to combine affine candidates.

Figure 13 illustrates an exemplary process for affine model creation from CPMVs from different candidates.

Figure 14 illustrates a block diagram of an embodiment of a video encoder.

Figure 15 illustrates a block diagram of an embodiment of a video decoder.

Figure 16 illustrates a block diagram of a system within which aspects of the present embodiments may be implemented.

DETAILED DESCRIPTION

Accordingly, one or more present embodiments aim to improve motion compensation in video encoding and decoding.

In the HEVC video compression standard, motion compensated temporal prediction is employed to exploit the redundancy that exists between successive pictures of a video.

A motion vector is associated to each prediction unit (PU), which we introduce now. Each CTU is represented by a Coding Tree in the compressed domain. This is a quad-tree division of the CTU, where each leaf is called a Coding Unit (CU), as shown in Figure 1.

Each CU is then given some Intra or Inter prediction parameters (Prediction Info). To do so, it is spatially partitioned into one or more Prediction Units (PUs), each PU being assigned some prediction information. The Intra or Inter coding mode is assigned on the CU level, as shown in Figure 2.

Exactly one motion vector is assigned to each PU in HEVC. This motion vector is used for motion compensated temporal prediction of the considered PU. Therefore, in HEVC, the motion model that links a predicted block and its reference block simply consists in a translation.

In an approach of the VTM (Versatile Test Model) developed by the JVET (Joint Video Exploration Team) group for VVC, some richer motion models are supported to improve temporal prediction. To do so, a PU can be spatially divided into sub-PU and a richer model can be used to assign each sub-PU a dedicated motion vector.

A CU is no longer divided into PU or TU, and some motion data are directly assigned to each CU. In this new codec design, a CU can be divided into sub-CU and a motion vector can be computed for each sub-CU.

One of the new motion models introduced in this approach is the affine model, which basically consists in using an affine motion model to represent the motion vectors in a CU.

The motion models used are illustrated in Figure 3 for 2 or 3 control points. The affine motion field for 2 control points consists in the following motion vector component values for each position (x, y) inside the considered block:

$$\begin{cases} v_x = \dfrac{(v_{1x} - v_{0x})}{w} x - \dfrac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \dfrac{(v_{1y} - v_{0y})}{w} x + \dfrac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases}$$

Equation 1: affine model used to generate the motion field inside a CU to predict

where (v_{0x}, v_{0y}) and (v_{1x}, v_{1y}) are the so-called control point motion vectors used to generate the affine motion field. (v_{0x}, v_{0y}) is the motion vector top-left corner control point. (v_{1x}, v_{1y}) is the motion vector top-right corner control point. A model with 3 control points is also considered.

In practice, to keep complexity reasonable, same motion vector is computed for each sample of 4x4 sub-block (sub-CU) of the considered CU, as illustrated in Figure 4. An affine motion vector is computed from the control point motion vectors, at the position of the center of each sub-block. The obtained motion vector (MV) is represented at 1/16-pel accuracy.

As a result, the prediction (PU) of a coding unit in the affine mode is built as motion compensating each sub-block with its own motion vector.

Affine motion compensation can be used in 2 ways in the VTM: Affine Inter (AF_INTER) and Affine Merge. They are introduced in the following.

Affine Inter (AF_INTER)

Figure 6 illustrates a process for creating new candidates in affine AMVP (Advanced Motion Vector Prediction) mode.

A CU in AMVP mode, whose size is larger than 8x8, can be predicted in Affine Inter mode. This is signaled through a flag in the bit-stream coded at CU level. The generation of the Affine Motion Field for that inter CU includes determining control point motion vectors (CPMV), which are obtained by the decoder through the addition of a motion vector difference plus a control point motion vector prediction (CPMVP). The CPMVP is a pair (for a 4-parameters affine model with 2 control points) or a triplet (for a 6-parameters affine model with 3 control points) of motion vector candidates which can be inherited from affine neighbors (as in the Affine Merge mode) or constructed from non-affine motion vectors respectively taken from the list (A, B, C) and (D, E) and/or (F, G), as illustrated in Figure 5.

First CPMVP are checked for validity using Equation, for a block of height H and width W:

$$\overrightarrow{\Delta Hor} = \overrightarrow{v1} - \overrightarrow{v0}$$

$$\overrightarrow{\Delta Ver} = \overrightarrow{v2} - \overrightarrow{v0}$$

$$DiffH = \frac{W}{2}$$

$$DiffV = \frac{H}{2}$$

$$validity = (\overrightarrow{\Delta Hor} != \vec{0} \;\&\&\; abs(\overrightarrow{\Delta Hor}.X) \leq DiffH \;\&\&\; abs(\overrightarrow{\Delta Hor}.Y) \leq DiffV \;\&\&$$
$$abs(\overrightarrow{\Delta Ver}.X) \leq DiffH \;\&\&\; abs(\overrightarrow{\Delta Ver}.Y) \leq DiffV)$$

Equation 2: Validity test for each CPMVP

Valid CPMVPs are then sorted depending on the value of a third motion vector $\overrightarrow{v_2}$, (taken from position F or G). The closest $\overrightarrow{v_2}$ is to the vector given by the affine motion model for the 4x4 sub-block at the same position as $\overrightarrow{v_2}$ is retained as the CPMVP.

For a block of height H and width W, the cost of each CPMVP is computed with Equation 3. In the following equation X and Y are respectively the horizontal and vertical components of a motion vector.

$$\overrightarrow{\Delta Hor} = \overrightarrow{v1} - \overrightarrow{v0}$$

$$\overrightarrow{\Delta Ver} = \overrightarrow{v2} - \overrightarrow{v0}$$

$$cost = abs(\overrightarrow{\Delta Hor}.X * H - \overrightarrow{\Delta Ver}.Y * W) + abs(\overrightarrow{\Delta Hor}.Y * H + \overrightarrow{\Delta Ver}.X * W)$$

Equation 3: Cost computed for each CPMVP

Affine Merge

In Affine Merge mode, a CU-level flag indicates if a CU in mode merge employs affine motion compensation. If so, then, the first available neighboring CU that has been coded in an affine mode is selected among the ordered list of candidate positions (A, B, C, D, E) shown in Figure 7.

Once the first neighboring CU in affine mode is obtained, then the 3 motion vectors $\overrightarrow{v_2}$, $\overrightarrow{v_3}$, and $\overrightarrow{v_4}$ from the top-left, top-right and bottom-left corners of the neighboring CU are retrieved (see Figure 8). Based on these three vectors, the two or three CPMVs of the top-left,

top-right and/or bottom-left corners of current CU are derived as follows (V0, V1, V2, V3 positions depicted in Figure 8:

$$\overrightarrow{v_0} = \overrightarrow{v_2} + (\overrightarrow{v_4} - \overrightarrow{v_2})\left(\frac{Y_{curr} - Y, V_{neighb}}{H_{neighb}}\right) + (\overrightarrow{v_3} - \overrightarrow{v_2})\left(\frac{X_{curr} - X_{neighb}}{W_{neighb}}\right)$$

$$\overrightarrow{v_1} = \overrightarrow{v_0} + (\overrightarrow{v_3} - \overrightarrow{v_2})\left(\frac{W_{curr}}{W_{neighb}}\right)$$

$$\overrightarrow{v_2} = \overrightarrow{v_0} + (\overrightarrow{v_4} - \overrightarrow{v_2})\left(\frac{H_{curr}}{H_{neighb}}\right)$$

Equation 4: derivation of current CU's CPMV based on the three corner motion vectors of the neighboring CU

When the control point motion vectors $\overrightarrow{v_0}$, $\overrightarrow{v_1}$ (and/or for the bottom left corner) of current CU are obtained, the motion field inside current CU is computed on a 4x4 sub-CU basis, through the model of Equation 1.

In the latest version of the VTM, more candidates for affine merge mode are considered. At the encoder, the best candidate is then selected through a Rate-Distortion Optimization process and the index of this best candidate is coded in the bitstream.

Another type of candidate is called temporal affine:

- Similarly to TMVP (Temporal Motion Vector Predictor) candidates, affine CUs are searched in reference images and added to the candidates list.

In another variant, a new process to create "virtual" candidates and add them to the list of candidates is described (as shown in Figure 9). The motivation is to create affine candidates when no affine CU is available around the current CU. To do so, an affine model is created by taking the motion of individual sub-block at the corner (as in the Affine AMVP mode).

In HEVC, the candidate list creation of the merge mode is enriched with combined candidates which creates new candidates with the L0 part of a candidate and the L1 part of a second candidate. Recently, in the VTM, those combined candidates have been replaced by pair-wise candidates which average two candidates. However, such combinations of candidates are not present in the affine coding modes.

Besides, as described previously, in the current design of VVC, an affine model can be of two different types: either a 4- or 6-parameters model which is described by 2 or 3 CPMVs

(Control Point Motion Vectors) respectively, i.e. (v_0, v_1) or (v_0, v_1, v_2). Thus, such combinations of candidates in the affine coding modes raises the issue of compatibility of the motion model.

Thus, it is desirable to enable combinations of already selected affine candidate(s). Accordingly, at least one embodiment of the present principles relates to constructing new affine candidates by combining already selected affine candidate(s). The at least one embodiment relates to an encoding/decoding method wherein motion estimation and/or motion compensation comprises:

- Defining combinations of affine candidates:
  - HEVC combined candidates for affine,
  - Pair-wise averaging for affine,
  - Affine model combinations of several or only one candidate,
  - Affine model averaging of several or only one candidate,
  - Cumulative combinations.
- Selecting a corresponding affine model type,
- Selecting a reference frame (if needed),
- Selecting the candidates to be combined.

First additional embodiment comprising adapting existing combinations for affine

Firstly, as mentioned above, HEVC already uses non-affine candidates' combinations in the merge process and such candidates can only be created in a bi-directional coding. They are the combination of the L0 part of one candidate and of the L1 part of a second candidate. The candidates used for the combination are picked from the already selected ones, by following a predefined list of combinations as {(0, 1) (1, 0), (0, 2), (2, 0), (1, 2), (2, 1), (0, 3), (3, 0), (1, 3), (3, 1), (2, 3), (3, 2)} where each pair represents the indexes of candidate used to get the L0 and L1 parts respectively (see Figure 10).

Secondly, in the latest VTM, pair-wise candidates have additionally supplemented the HEVC combined candidates in the merge process.

Figure 11 illustrates an exemplary pair-wise candidates generation process 11. These candidates are generated by averaging predefined pairs of candidates of the current merge candidate list (921). The predefined pairs are {(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)} where each pair represents the indexes of the candidates to be averaged (922, 923). The averaged

motion vectors are calculated separately for each reference list. If both motion vectors are available in one list, these two motion vectors are averaged even when they point to different reference pictures; if only one motion vector is available, use the one directly; if no motion vector is available, keep this list invalid (923-925).

In the affine process, as there are now several candidates in competition in the merge and AMVP processes, if the list of affine candidates is not full, then it can be filled with combinations of candidates as in the merge process.

Adapting HEVC combination for affine

Figure 12 illustrates an exemplary process 12 to combine affine candidates.

As in HEVC, it is possible to construct new candidates by combining the L0 part of one candidate and the L1 part of a second one, where Lx part is the corresponding affine model described by its CPMVs (1202-1206). This construction may follow similar constraints as for the HEVC merge process (see, e.g., Figure 12):

(i) it must be in a bi-directional case (1202),

(ii) both candidates must exist in the list (yes, 1203),

(iii) the first one must hold a L0 affine model (yes, 1204) and

(iv) the second a L1 affine model different from the L1 part of the first candidate or pointing to a different reference frame than the L1 part of the first candidate (yes, 1205).

The combined candidates can be selected from a predefined list of combinations as, for example, like in HEVC in {(0, 1) (1, 0), (0, 2), (2, 0), (1, 2), (2, 1), (0, 3), (3, 0), (1, 3), (3, 1), (2, 3), (3, 2)} where each pair represents the indexes of candidate used to get the L0 and L1 parts respectively. However, it can also be in any other predefined list of combinations, and in any order.

Adapting pair-wise averaging for affine

Some pair-wise candidates can also be constructed in the affine modes. In that case, instead of averaging one motion vector per reference list, several motion vectors (2 or 3 for 4- or 6-parameters affine model resp.) must be averaged in each reference list. If both set of CPMVs are available in one list, these two sets are averaged even when they point to different reference pictures; if only one set of CPMVs is available, use the one directly; if no set of CPMVs is available, keep this list invalid.

The averaged candidates can be selected from predefined pairs as, for example, in {(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)} where each pair represents the indexes of the candidates to be averaged. But it can also be from any other predefined pairs, and in any order.

This averaging can also be defined for tuple of candidates, or more candidates.

We implemented and tested the above pair-wise adaptation for affine. The results were compared with the results obtained by VTM 6.0. The BD-rate performance improvements are shown in Table 1 below.

	Random Access Main 10				
	Over VTM-6.0				
	Y	U	V	EncT	DecT
Class A1	-0.01%	0.06%	-0.10%	100%	100%
Class A2	0.01%	0.00%	0.06%	101%	101%
Class B	-0.02%	0.09%	-0.08%	98%	97%
Class C	-0.02%	-0.04%	-0.07%	99%	99%
Overall	-0.01%	0.03%	-0.05%	99%	99%
Class D	-0.06%	-0.02%	-0.04%	98%	99%
Class F (optional)	-0.02%	-0.04%	-0.06%	100%	101%

	Low delay B Main10				
	Over VTM-6.0				
	Y	U	V	EncT	DecT
Class B	-0.02%	-0.02%	-0.17%	102%	101%
Class C	-0.08%	-0.28%	-0.05%	99%	99%
Class E	-0.07%	0.17%	-0.11%	99%	100%
Overall	-0.05%	-0.06%	-0.12%	100%	100%
Class D	-0.06%	-0.10%	-0.29%	99%	100%
Class F (optional)	0.00%	-0.05%	0.47%	100%	101%

Table 1: Improved BD-rate performance of the proposed pair-wise adaptation for affine compared to VTM 6.0

Second additional embodiment comprising combining affine models

Instead of combining the candidates at the reference list level, the affine model can be combined at the CPMV level.

First variant of second embodiment: affine models' combinations with several candidates

By using several already selected candidates, and for each reference list independently, it is possible to combine the affine models by combining the CPMVs in order to create new affine models. Figure 13 illustrates an exemplary process 13 for affine model creation from CPMVs from different candidates.

By considering, for example, a pair of candidates:

- If both candidates use a 4-parameters affine model ((v_0, v_1) and (v_0', v_1')), the new affine model is also a 4-parameters affine model constructed with v_0 coming from the first candidate and v_1' coming from the second one. It can only be constructed if v_1' is different from v_1 or if the L1 reference frame is different from the L0 one (see, e.g., Figure 13).
- If both candidates use a 6-parameters affine model ((v_0, v_1, v_2) and (v_0', v_1', v_2')), the new affine model can be:
  - Restricted to a 4-parameters affine model constructed with (v_0, v_1'). It can only be constructed if v_1' is different from v_1 or if the L1 reference frame is different from the L0 one,
  - Or a 6-parameters affine model constructed with (a) (v_0, v_1', v_2), (b) (v_0, v_1, v_2') or (c) (v_0, v_1', v_2'). They can only be constructed:
    - For (a), if v_1' is different from v_1 or if the L1 reference frame is different from the L0 one.
    - For (b), if v_2' is different from v_2 or if the L1 reference frame is different from the L0 one.
    - For (c), if both above conditions are fulfilled.
- If one uses a 4-parameters and the other a 6-parameters affine model, with the 4-parameters one as (i) the first one ((v_0, v_1) and (v_0', v_1', v_2')) or (ii) the second one ((v_0, v_1, v_2) and (v_0', v_1')), the new affine model can be:
  - Restricted to a 4-parameters affine model constructed with (v_0, v_1'). It can only be constructed if v_1' is different from v_1 or if the L1 reference frame is different from the L0 one,
  - Or enlarged to a 6-parameters affine model constructed with (i) (v_0, v_1, v_2') or (v_0, v_1', v_2') or (ii) (v_0, v_1', v_2). They can only be constructed if v_1' is different from v_1 or if the L1 reference frame is different from the L0 one for the two last ones, only the first (i) can always be constructed.

These model combined candidates can be selected from a predefined list of combinations as, for example, like in HEVC in {(0, 1) (1, 0), (0, 2), (2, 0), (1, 2), (2, 1), (0, 3), (3, 0), (1, 3), (3, 1), (2, 3), (3, 2)} where each pair represents the indexes of candidate used. But it can also be in any other predefined list of combinations, and in any order.

Such model combinations can also be defined, in the same way, for tuple of candidates, since affine models can be defined with up to 3 CPMVs.

Second variant of second embodiment affine model combinations with only one candidate

In the case there is only one candidate in the current constructed list, a case where adding some candidates can be very useful, none of the above affine model combinations can apply. But some combinations are still possible by using only one candidate:

- If the candidate is bi-directional, then two uni-directional candidates can be derived.
- If the candidate is bi-directional, then the CPMVs from L0 and L1 can be mixed:
  - If the candidate uses a 4-parameters affine model ((v_0, v_1) L0 and (v_0', v_1') L1), the new affine model is also a 4-parameters affine model constructed with:
    - one change: {(v_0, v_1) (v_0', v_1)}, {(v_0, v_1) (v_0, v_1')}, {(v_0, v_1') (v_0', v_1')}, {(v_0', v_1) (v_0', v_1')},
    - or two changes: {(v_0, v_1'} (v_0', v_1)}, {(v_0, v_1'} (v_0, v_1')}, {(v_0', v_1) (v_0, v_1')}, {(v_0', v_1) (v_0', v_1)} , {(v_0, v_1) (v_0, v_1)}, {(v_0', v_1') (v_0', v_1')}.

They can only be constructed if the modified CPMV(s) is (are) different from original CPMV(s) or if the L1 reference frame is different from the L0 one.

- If the candidate uses a 6-parameters affine model ((v_0, v_1, v_2) L0 and (v_0', v_1', v_2') L1), the new affine model can be:
  - Restricted to a 4-parameters affine model constructed as presented in the previous bullet,
  - Or a 6-parameters affine model constructed with one, two or three changes on the same principles as for 4-parameters. They can only be constructed if the modified CPMV(s) is (are) different from original CPMV(s) or if the L1 reference frame is different from the L0 one.
- Whether the candidate is uni- or bi-directional, for each reference list independently, the CPMVs can be swapped:
- If the candidate uses a 4-parameters affine model (v_0, v_1), the new affine model is also a 4-parameters affine model constructed with one change (v_1, v_0). It can only be constructed if v_0 and v_1 are different.

- If the candidate uses a 6-parameters affine model (v_0, v_1, v_2), the new affine model can be:
  - Restricted to a 4-parameters affine model constructed with (v_0, v_1), (v_0, v_2), (v_1, v_0), (v_1, v_2), (v_2, v_0) or (v_2, v_1). It can only be constructed if v_0, v_1 and v_2 are different.
  - Or a 6-parameters affine model constructed with (v_0, v_2, v_1), (v_1, v_0, v_2), (v_1, v_2, v_0), (v_2, v_0, v_1) or (v_2, v_1, v_0). It can only be constructed if v_0, v_1 and v_2 are different.

Third embodiment comprising averaging affine models

In the pair-wise averaging for affine presented previously, for each reference list independently, the candidates' affine models are averaged, i.e. each CPMV of the first candidate is averaged with the corresponding CPMV of the second candidate. Instead of averaging models of several candidates, the affine model can be averaged at the CPMV level.

First variant of the third embodiment: affine models' averaging with several candidates

Since several CPMVs are involved in an affine model, it is possible to average only 1 or 2 CPMV(s):

- If both candidates use a 4-parameters affine model ((v_0, v_1) and (v_0', v_1')), the new affine model is also a 4-parameters affine model constructed with (avg(v_0, v_0'), v_1), (avg(v_0, v_0'), v_1'), (v_0, avg(v_1, v_1')) or (v_0', avg(v_1, v_1')). They can only be constructed if v_0' is different from v_0 for the two first ones, and if v_1' is different from v_1 for the two last ones.
- If both candidates use a 6-parameters affine model ((v_0, v_1, v_2) and (v_0', v_1', v_2')), the new affine model can be:
- Restricted to a 4-parameters affine model constructed as presented in the previous bullet,
- Or a 6-parameters affine model constructed with one or two averages based on the same principles as for 4-parameters.
- If one uses a 4-parameters and the other a 6-parameters affine model, with the 4-parameters one as (i) the first one ((v_0, v_1) and (v_0', v_1', v_2')) or (ii) the second one ((v_0, v_1, v_2) and (v_0', v_1')), the new affine model can be:
- Restricted to a 4-parameters affine model constructed as presented in the first bullet,

- Or enlarged to a 6-parameters affine model constructed as a 4-parameters affine model enriched with v_2 or v_2' depending on which is available.

These model averaged candidates can be selected from a predefined list of combinations as, for example, like in HEVC in {(0, 1) (1, 0), (0, 2), (2, 0), (1, 2), (2, 1), (0, 3), (3, 0), (1, 3), (3, 1), (2, 3), (3, 2)} where each pair represents the indexes of candidate used. But it can also be in any other predefined list of combinations, and in any order.

Advantageously, this embodiment allows to give a different priority for the CPMV v_0, with respect to v_1 and v_2 in the combination for instance when using (v_0, avg(v_1, v_1')) or (v_0', avg(v_1, v_1')) since the motion at v_0 (or v_0') being the center of rotational model is more reliable.

This affine models' averaging can also be defined for tuple of candidates, or more candidates.

Second variant of third embodiment: affine model averaging with only one candidate

It is also possible to average CPMV(s) of only one candidate:

- If the candidate is bi-directional, then a uni-directional candidate can be constructed by averaging all CPMVs of both lists.
- If the candidate is bi-directional, then the CPMVs from L0 and L1 can be averaged:
- If the candidate uses a 4-parameters affine model ((v_0, v_1) L0 and (v_0', v_1') L1), the new affine model is also a 4-parameters affine model constructed with:
  - one average: {(v_0, v_1) (v_0', avg(v_1, v_1'))}, {(v_0, v_1) (avg(v_0, v_0'), v_1')}, {(v_0, avg(v_1, v_1')) (v_0', v_1')}, {(avg(v_0, v_0'), v_1) (v_0', v_1')},
  - or two averages: {(v_0, v_1) (avg(v_0, v_0'), avg(v_1, v_1'))}, {(avg(v_0, v_0'), avg(v_1, v_1')) (v_0', v_1')}, {(v_0, avg(v_1, v_1')) (v_0', avg(v_1, v_1'))}, {(v_0, avg(v_1, v_1')) (avg(v_0, v_0'), v_1')}, {(avg(v_0, v_0'), v_1) (avg(v_0, v_0'), v_1')}, {(avg(v_0, v_0'), v_1) (v_0', avg(v_1, v_1'))}.
- If the candidate uses a 6-parameters affine model ((v_0, v_1, v_2) L0 and (v_0', v_1', v_2') L1), the new affine model can be:
  - Restricted to a 4-parameters affine model constructed as presented in the previous bullet,
  - Or a 6-parameters affine model constructed with one or two averages based on the same principles as for 4-parameters.

- Whether the candidate is uni- or bi-directional, for each reference list independently, the CPMVs can be averaged:
- If the candidate uses a 4-parameters affine model (v_0, v_1), the new affine model is also a 4-parameters affine model constructed with one average (v_0, avg(v_0, v_1)) or (avg(v_0, v_1), v_1). It can only be constructed if v_0 and v_1 are different.
- If the candidate uses a 6-parameters affine model (v_0, v_1, v_2), the new affine model can be:
  - Restricted to a 4-parameters affine model constructed with one average (v_0, avg(v_1, v_2)), (v_0, avg(v_0, v_1)), (v_0, avg(v_0, v_2)), (avg(v_0, v_2), v_1), (avg(v_0, v_1), v_1) or (avg(v_1, v_2), v_1). It can only be constructed if v_0, v_1 and v_2 are different.
  - Or a 6-parameters affine model constructed with one or two averages based on the same principles as for 4-parameters. It can only be constructed if v_0, v_1 and v_2 are different.

Advantageously, this embodiment allows to increase the number of candidates even when only one candidate is available.

Fourth embodiment coupling affine models' combinations and averaging

Several proposed combinations are performed for each reference list independently, it is then possible to couple two of these combinations, one for each list.

Furthermore, several proposed combinations are performed for different CPMVs, it is then possible to mix several of them. For example, one can use a model combination that changes v_1 and/or v_2, but takes an average of the v_0s.

Fifth embodiment comprising a selection of affine model type

Since the affine model type, as the number of parameters of the affine model, is signaled at the CU level, combinations and averaging require to select the affine model type for the whole CU.

According to non-limiting examples, a selection of affine model type comprises one of:

- Restrict to 4-parameters affine model whatever the candidates' type, i.e. in both lists the v_2 CPMV, if it exists, is ignored, all combinations or averaging are only performed on v_0 and v_1.

- Restrict to 6-parameters affine model whatever the candidates' type, i.e. in both lists if the v_2 CPMV do not exist, it is calculated for the two other CPMVs using Equation 1, all combinations or averaging are then performed on v_0, v_1 and v_2.
- Use the minimum number of parameters hold by combined or averaged candidates, i.e. if at least one candidate uses a 4-parameters affine model, then the generated candidate uses also a 4-parameters affine model; it can only use a 6-parameters affine model if all candidates use a 6-parameters affine model.
- Use the maximum number of parameters hold by combined or averaged candidates, i.e. if at least one candidate uses a 6-parameters affine model, then the generated candidate uses also a 6-parameters affine model; it can only use a 4-parameters affine model if all candidates use a 4-parameters affine model.
- When the order of the candidates used for combination is important, as for HEVC combined and models' combinations and averaging, it then possible to:
  - Use the same number of parameters as the first candidate used for combination,
  - Use the same number of parameters as the last candidate used for combination.

The chosen selection process can be predefined in the standard, linked to a profile, or signaled at the SPS, PPS, slice or CU level.

Sixth embodiment comprising reference frame selection

In all the cases, if a reference frame is needed (merge case), then according to non-limiting examples, the combined candidate (HEVC combined, pair-wise averaged or model combinations) can use:

- The same reference frame as the first candidate used for combination,
- The smallest reference frame index held by all the candidates used for combination,
- The most used reference frame by all the candidates used for combination,
- The reference frame with the smallest POC between all the candidates used for combination,
- The first reference frame in the list,

According to another variant, the firth embodiment further comprises scaling candidates with respect to the POC of the reference frame of the candidate with respect to the chosen reference frame.

Seventh embodiment comprising candidate selection for combination

In all the cases, if the combined candidate (HEVC combined, pair-wise averaged or model combinations) is constructed from more than one already selected one, then it can use:

- Any of the already selected candidates (i.e. combine all),
- Only already selected inherited candidates (i.e. combine only inherited),
- Only already selected virtual candidates (i.e. combine only virtual),
- Only already selected inherited and virtual candidates (i.e. combine only inherited with virtual),
- Depending on the number of combined candidates to be inserted in the list, several of the above usages can be used together with different order. For example, if two combined candidates are allowed, then it can be constructed (i) from inherited (if at least 2 inherited candidates are already selected), then (ii) from virtual (if at least 2 virtual candidates are already selected), and (iii) from inherited-virtual (if at least 1 inherited and 1 virtual candidate are already selected).

This application describes a variety of aspects, including tools, features, embodiments, models, approaches, etc. Many of these aspects are described with specificity and, at least to show the individual characteristics, are often described in a manner that may sound limiting. However, this is for purposes of clarity in description, and does not limit the application or scope of those aspects. Indeed, all of the different aspects can be combined and interchanged to provide further aspects. Moreover, the aspects can be combined and interchanged with aspects described in earlier filings as well.

The aspects described and contemplated in this application can be implemented in many different forms. Figures 14, 15 and 16 below provide some embodiments, but other embodiments are contemplated and the discussions of Figures 14, 15 and 16 do not limit the breadth of the implementations. At least one of the aspects generally relates to video encoding and decoding, and at least one other aspect generally relates to transmitting a bitstream generated or encoded. These and other aspects can be implemented as a method, an apparatus, a computer readable storage medium having stored thereon instructions for encoding or decoding video data according to any of the methods described, and/or a computer readable storage medium having stored thereon a bitstream generated according to any of the methods described.

In the present application, the terms "reconstructed" and "decoded" may be used interchangeably, the terms "pixel" and "sample" may be used interchangeably, the terms "image," "picture" and "frame" may be used interchangeably. Usually, but not necessarily, the term "reconstructed" is used at the encoder side while "decoded" is used at the decoder side.

Various methods are described herein, and each of the methods comprises one or more steps or actions for achieving the described method. Unless a specific order of steps or actions is required for proper operation of the method, the order and/or use of specific steps and/or actions may be modified or combined.

Various methods and other aspects described in this application can be used to modify modules, for example, the motion estimation and/or motion compensation (175, 170, 275), of a video encoder 100 and decoder 200 as shown in Figure 14 and Figure 15. Moreover, the present aspects are not limited to VVC or HEVC, and can be applied, for example, to other standards and recommendations, whether pre-existing or future-developed, and extensions of any such standards and recommendations (including VVC and HEVC). Unless indicated otherwise, or technically precluded, the aspects described in this application can be used individually or in combination.

Various numeric values are used in the present application, for example, the number of candidates or the number of motion vector in the motion model. The specific values are for example purposes and the aspects described are not limited to these specific values.

Figure 14 illustrates an encoder 100. Variations of this encoder 100 are contemplated, but the encoder 100 is described below for purposes of clarity without describing all expected variations.

Before being encoded, the video sequence may go through pre-encoding processing (101), for example, applying a color transform to the input color picture (e.g., conversion from RGB 4:4:4 to YCbCr 4:2:0), or performing a remapping of the input picture components in order to get a signal distribution more resilient to compression (for instance using a histogram equalization of one of the color components). Metadata can be associated with the pre-processing, and attached to the bitstream.

In the encoder 100, a picture is encoded by the encoder elements as described below. The picture to be encoded is partitioned (102) and processed in units of, for example, CUs. Each unit is encoded using, for example, either an intra or inter mode. When a unit is encoded

in an intra mode, it performs intra prediction (160). In an inter mode, motion estimation (175) and compensation (170) are performed. The encoder decides (105) which one of the intra mode or inter mode to use for encoding the unit, and indicates the intra/inter decision by, for example, a prediction mode flag. Prediction residuals are calculated, for example, by subtracting (110) the predicted block from the original image block.

The prediction residuals are then transformed (125) and quantized (130). The quantized transform coefficients, as well as motion vectors and other syntax elements, are entropy coded (145) to output a bitstream. The encoder can skip the transform and apply quantization directly to the non-transformed residual signal. The encoder can bypass both transform and quantization, i.e., the residual is coded directly without the application of the transform or quantization processes.

The encoder decodes an encoded block to provide a reference for further predictions. The quantized transform coefficients are de-quantized (140) and inverse transformed (150) to decode prediction residuals. Combining (155) the decoded prediction residuals and the predicted block, an image block is reconstructed. In-loop filters (165) are applied to the reconstructed picture to perform, for example, deblocking/SAO (Sample Adaptive Offset) filtering to reduce encoding artifacts. The filtered image is stored at a reference picture buffer (180).

Figure 15 illustrates a block diagram of a video decoder 200. In the decoder 200, a bitstream is decoded by the decoder elements as described below. Video decoder 200 generally performs a decoding pass reciprocal to the encoding pass as described in Figure 14. The encoder 100 also generally performs video decoding as part of encoding video data.

In particular, the input of the decoder includes a video bitstream, which can be generated by video encoder 100. The bitstream is first entropy decoded (230) to obtain transform coefficients, motion vectors, and other coded information. The picture partition information indicates how the picture is partitioned. The decoder may therefore divide (235) the picture according to the decoded picture partitioning information. The transform coefficients are de-quantized (240) and inverse transformed (250) to decode the prediction residuals. Combining (255) the decoded prediction residuals and the predicted block, an image block is reconstructed. The predicted block can be obtained (270) from intra prediction (260) or motion-compensated prediction (i.e., inter prediction) (275). In-loop filters (265) are applied to the reconstructed image. The filtered image is stored at a reference picture buffer (280).

The decoded picture can further go through post-decoding processing (285), for example, an inverse color transform (e.g. conversion from YCbCr 4:2:0 to RGB 4:4:4) or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (101). The post-decoding processing can use metadata derived in the pre-encoding processing and signaled in the bitstream.

Figure 16 illustrates a block diagram of an example of a system in which various aspects and embodiments are implemented. System 1000 can be embodied as a device including the various components described below and is configured to perform one or more of the aspects described in this document. Examples of such devices, include, but are not limited to, various electronic devices such as personal computers, laptop computers, smartphones, tablet computers, digital multimedia set top boxes, digital television receivers, personal video recording systems, connected home appliances, and servers. Elements of system 1000, singly or in combination, can be embodied in a single integrated circuit (IC), multiple ICs, and/or discrete components. For example, in at least one embodiment, the processing and encoder/decoder elements of system 1000 are distributed across multiple ICs and/or discrete components. In various embodiments, the system 1000 is communicatively coupled to one or more other systems, or other electronic devices, via, for example, a communications bus or through dedicated input and/or output ports. In various embodiments, the system 1000 is configured to implement one or more of the aspects described in this document.

The system 1000 includes at least one processor 1010 configured to execute instructions loaded therein for implementing, for example, the various aspects described in this document. Processor 1010 can include embedded memory, input output interface, and various other circuitries as known in the art. The system 1000 includes at least one memory 1020 (e.g., a volatile memory device, and/or a non-volatile memory device). System 1000 includes a storage device 1040, which can include non-volatile memory and/or volatile memory, including, but not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), flash, magnetic disk drive, and/or optical disk drive. The storage device 1040 can include an internal storage device, an attached storage device (including detachable and non-detachable storage devices), and/or a network accessible storage device, as non-limiting examples.

System 1000 includes an encoder/decoder module 1030 configured, for example, to process data to provide an encoded video or decoded video, and the encoder/decoder module 1030 can include its own processor and memory. The encoder/decoder module 1030 represents module(s) that can be included in a device to perform the encoding and/or decoding functions. As is known, a device can include one or both of the encoding and decoding modules. Additionally, encoder/decoder module 1030 can be implemented as a separate element of system 1000 or can be incorporated within processor 1010 as a combination of hardware and software as known to those skilled in the art.

Program code to be loaded onto processor 1010 or encoder/decoder 1030 to perform the various aspects described in this document can be stored in storage device 1040 and subsequently loaded onto memory 1020 for execution by processor 1010. In accordance with various embodiments, one or more of processor 1010, memory 1020, storage device 1040, and encoder/decoder module 1030 can store one or more of various items during the performance of the processes described in this document. Such stored items can include, but are not limited to, the input video, the decoded video or portions of the decoded video, the bitstream, matrices, variables, and intermediate or final results from the processing of equations, formulas, operations, and operational logic.

In some embodiments, memory inside of the processor 1010 and/or the encoder/decoder module 1030 is used to store instructions and to provide working memory for processing that is needed during encoding or decoding. In other embodiments, however, a memory external to the processing device (for example, the processing device can be either the processor 1010 or the encoder/decoder module 1030) is used for one or more of these functions. The external memory can be the memory 1020 and/or the storage device 1040, for example, a dynamic volatile memory and/or a non-volatile flash memory. In several embodiments, an external non-volatile flash memory is used to store the operating system of, for example, a television. In at least one embodiment, a fast external dynamic volatile memory such as a RAM is used as working memory for video coding and decoding operations, such as for MPEG-2 (MPEG refers to the Moving Picture Experts Group, MPEG-2 is also referred to as ISO/IEC 13818, and 13818-1 is also known as H.222, and 13818-2 is also known as H.262), HEVC (HEVC refers to High Efficiency Video Coding, also known as H.265 and MPEG-H Part 2), or VVC (Versatile Video Coding, a new standard being developed by JVET, the Joint Video Experts Team).

The input to the elements of system 1000 can be provided through various input devices as indicated in block 1130. Such input devices include, but are not limited to, (i) a radio frequency (RF) portion that receives an RF signal transmitted, for example, over the air by a broadcaster, (ii) a Component (COMP) input terminal (or a set of COMP input terminals), (iii) a Universal Serial Bus (USB) input terminal, and/or (iv) a High *Definition* Multimedia Interface (HDMI) input terminal. Other examples, not shown in Figure 16, include composite video.

In various embodiments, the input devices of block 1130 have associated respective input processing elements as known in the art. For example, the RF portion can be associated with elements suitable for (i) selecting a desired frequency (also referred to as selecting a signal, or band-limiting a signal to a band of frequencies), (ii) downconverting the selected signal, (iii) band-limiting again to a narrower band of frequencies to select (for example) a signal frequency band which can be referred to as a channel in certain embodiments, (iv) demodulating the downconverted and band-limited signal, (v) performing error correction, and (vi) demultiplexing to select the desired stream of data packets. The RF portion of various embodiments includes one or more elements to perform these functions, for example, frequency selectors, signal selectors, band-limiters, channel selectors, filters, downconverters, demodulators, error correctors, and demultiplexers. The RF portion can include a tuner that performs various of these functions, including, for example, downconverting the received signal to a lower frequency (for example, an intermediate frequency or a near-baseband frequency) or to baseband. In one set-top box embodiment, the RF portion and its associated input processing element receives an RF signal transmitted over a wired (for example, cable) medium, and performs frequency selection by filtering, downconverting, and filtering again to a desired frequency band. Various embodiments rearrange the order of the above-described (and other) elements, remove some of these elements, and/or add other elements performing similar or different functions. Adding elements can include inserting elements in between existing elements, such as, for example, inserting amplifiers and an analog-to-digital converter. In various embodiments, the RF portion includes an antenna.

Additionally, the USB and/or HDMI terminals can include respective interface processors for connecting system 1000 to other electronic devices across USB and/or HDMI connections. It is to be understood that various aspects of input processing, for example, Reed-Solomon error correction, can be implemented, for example, within a separate input processing

IC or within processor 1010 as necessary. Similarly, aspects of USB or HDMI interface processing can be implemented within separate interface ICs or within processor 1010 as necessary. The demodulated, error corrected, and demultiplexed stream is provided to various processing elements, including, for example, processor 1010, and encoder/decoder 1030 operating in combination with the memory and storage elements to process the datastream as necessary for presentation on an output device.

Various elements of system 1000 can be provided within an integrated housing, within the integrated housing, the various elements can be interconnected and transmit data therebetween using suitable connection arrangement, for example, an internal bus as known in the art, including the Inter-IC (I2C) bus, wiring, and printed circuit boards.

The system 1000 includes communication interface 1050 that enables communication with other devices via communication channel 1060. The communication interface 1050 can include, but is not limited to, a transceiver configured to transmit and to receive data over communication channel 1060. The communication interface 1050 can include, but is not limited to, a modem or network card and the communication channel 1060 can be implemented, for example, within a wired and/or a wireless medium.

Data is streamed, or otherwise provided, to the system 1000, in various embodiments, using a wireless network such as a Wi-Fi network, for example IEEE 802.11 (IEEE refers to the Institute of Electrical and Electronics Engineers). The Wi-Fi signal of these embodiments is received over the communications channel 1060 and the communications interface 1050 which are adapted for Wi-Fi communications. The communications channel 1060 of these embodiments is typically connected to an access point or router that provides access to external networks including the Internet for allowing streaming applications and other over-the-top communications. Other embodiments provide streamed data to the system 1000 using a set-top box that delivers the data over the HDMI connection of the input block 1130. Still other embodiments provide streamed data to the system 1000 using the RF connection of the input block 1130. As indicated above, various embodiments provide data in a non-streaming manner. Additionally, various embodiments use wireless networks other than Wi-Fi, for example a cellular network or a Bluetooth network.

The system 1000 can provide an output signal to various output devices, including a display 1100, speakers 1110, and other peripheral devices 1120. The display 1100 of various embodiments includes one or more of, for example, a touchscreen display, an organic light-

emitting diode (OLED) display, a curved display, and/or a foldable display. The display 1100 can be for a television, a tablet, a laptop, a cell phone (mobile phone), or other device. The display 1100 can also be integrated with other components (for example, as in a smart phone), or separate (for example, an external monitor for a laptop). The other peripheral devices 1120 include, in various examples of embodiments, one or more of a stand-alone digital video disc (or digital versatile disc) (DVR, for both terms), a disk player, a stereo system, and/or a lighting system. Various embodiments use one or more peripheral devices 1120 that provide a function based on the output of the system 1000. For example, a disk player performs the function of playing the output of the system 1000.

In various embodiments, control signals are communicated between the system 1000 and the display 1100, speakers 1110, or other peripheral devices 1120 using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications protocols that enable device-to-device control with or without user intervention. The output devices can be communicatively coupled to system 1000 via dedicated connections through respective interfaces 1070, 1080, and 1090. Alternatively, the output devices can be connected to system 1000 using the communications channel 1060 via the communications interface 1050. The display 1100 and speakers 1110 can be integrated in a single unit with the other components of system 1000 in an electronic device such as, for example, a television. In various embodiments, the display interface 1070 includes a display driver, such as, for example, a timing controller (T Con) chip.

The display 1100 and speaker 1110 can alternatively be separate from one or more of the other components, for example, if the RF portion of input 1130 is part of a separate set-top box. In various embodiments in which the display 1100 and speakers 1110 are external components, the output signal can be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

The embodiments can be carried out by computer software implemented by the processor 1010 or by hardware, or by a combination of hardware and software. As a non-limiting example, the embodiments can be implemented by one or more integrated circuits. The memory 1020 can be of any type appropriate to the technical environment and can be implemented using any appropriate data storage technology, such as optical memory devices, magnetic memory devices, semiconductor-based memory devices, fixed memory, and removable memory, as non-limiting examples. The processor 1010 can be of any type

appropriate to the technical environment, and can encompass one or more of microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

Various implementations involve decoding. "Decoding", as used in this application, can encompass all or part of the processes performed, for example, on a received encoded sequence in order to produce a final output suitable for display. In various embodiments, such processes include one or more of the processes typically performed by a decoder, for example, entropy decoding, inverse quantization, inverse transformation, and differential decoding. In various embodiments, such processes also, or alternatively, include processes performed by a decoder of various implementations described in this application, for example, motion compensation comprising determining a candidate for affine motion compensation of a current block by combining at least one previously determined affine candidate.

As further examples, in one embodiment "decoding" refers only to entropy decoding, in another embodiment "decoding" refers only to differential decoding, and in another embodiment "decoding" refers to a combination of entropy decoding and differential decoding. Whether the phrase "decoding process" is intended to refer specifically to a subset of operations or generally to the broader decoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

Various implementations involve encoding. In an analogous way to the above discussion about "decoding", "encoding" as used in this application can encompass all or part of the processes performed, for example, on an input video sequence in order to produce an encoded bitstream. In various embodiments, such processes include one or more of the processes typically performed by an encoder, for example, partitioning, differential encoding, transformation, quantization, and entropy encoding. In various embodiments, such processes also, or alternatively, include processes performed by an encoder of various implementations described in this application, for example, motion compensation comprising determining a candidate for affine motion compensation of a current block by combining at least one previously determined affine candidate.

As further examples, in one embodiment "encoding" refers only to entropy encoding, in another embodiment "encoding" refers only to differential encoding, and in another embodiment "encoding" refers to a combination of differential encoding and entropy encoding. Whether the phrase "encoding process" is intended to refer specifically to a subset of operations

or generally to the broader encoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

Note that the syntax elements as used herein are descriptive terms. As such, they do not preclude the use of other syntax element names.

When a figure is presented as a flow diagram, it should be understood that it also provides a block diagram of a corresponding apparatus. Similarly, when a figure is presented as a block diagram, it should be understood that it also provides a flow diagram of a corresponding method/process.

The implementations and aspects described herein can be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed can also be implemented in other forms (for example, an apparatus or program). An apparatus can be implemented in, for example, appropriate hardware, software, and firmware. The methods can be implemented in, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants ("PDAs"), and other devices that facilitate communication of information between end-users.

Reference to “one embodiment” or “an embodiment” or “one implementation” or “an implementation”, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrase “in one embodiment” or “in an embodiment” or “in one implementation” or “in an implementation”, as well any other variations, appearing in various places throughout this application are not necessarily all referring to the same embodiment.

Additionally, this application may refer to “determining” various pieces of information. Determining the information can include one or more of, for example, estimating the information, calculating the information, predicting the information, or retrieving the information from memory.

Further, this application may refer to “accessing” various pieces of information.

Accessing the information can include one or more of, for example, receiving the information, retrieving the information (for example, from memory), storing the information, moving the information, copying the information, calculating the information, determining the information, predicting the information, or estimating the information.

Additionally, this application may refer to "receiving" various pieces of information. Receiving is, as with "accessing", intended to be a broad term. Receiving the information can include one or more of, for example, accessing the information, or retrieving the information (for example, from memory). Further, "receiving" is typically involved, in one way or another, during operations such as, for example, storing the information, processing the information, transmitting the information, moving the information, copying the information, erasing the information, calculating the information, determining the information, predicting the information, or estimating the information.

It is to be appreciated that the use of any of the following "/", "and/or", and "at least one of", for example, in the cases of "A/B", "A and/or B" and "at least one of A and B", is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of "A, B, and/or C" and "at least one of A, B, and C", such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as is clear to one of ordinary skill in this and related arts, for as many items as are listed.

Also, as used herein, the word "signal" refers to, among other things, indicating something to a corresponding decoder. For example, in certain embodiments the encoder signals a particular one of a plurality of motion model (4- or 6-paramters). In this way, in an embodiment the same parameter is used at both the encoder side and the decoder side. Thus, for example, an encoder can transmit (explicit signaling) a particular parameter to the decoder so that the decoder can use the same particular parameter. Conversely, if the decoder already has the particular parameter as well as others, then signaling can be used without transmitting (implicit signaling) to simply allow the decoder to know and select the particular parameter. By avoiding transmission of any actual functions, a bit savings is realized in various

embodiments. It is to be appreciated that signaling can be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth are used to signal information to a corresponding decoder in various embodiments. While the preceding relates to the verb form of the word "signal", the word "signal" can also be used herein as a noun.

As will be evident to one of ordinary skill in the art, implementations can produce a variety of signals formatted to carry information that can be, for example, stored or transmitted. The information can include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal can be formatted to carry the bitstream of a described embodiment. Such a signal can be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting can include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries can be, for example, analog or digital information. The signal can be transmitted over a variety of different wired or wireless links, as is known. The signal can be stored on a processor-readable medium.

We describe a number of embodiments. Features of these embodiments can be provided alone or in any combination. Further, embodiments can include one or more of the following features, devices, or aspects, alone or in any combination, across various claim categories and types:

- Combining candidates by selecting motion vectors/models among candidates for affine motion compensation applied in the decoder and/or encoder,
- Combining candidates by pair-wise averaging motion vectors/models for affine motion compensation applied in the decoder and/or encoder,
- Combining affine model of several or only one candidate for affine motion compensation applied in the decoder and/or encoder,
- Averaging affine model of several or only one candidate for affine motion compensation applied in the decoder and/or encoder,
- Any combination according to any of the embodiments described with combining/averaging of candidates,
- Selecting a corresponding affine model type for affine combination applied in the decoder and/or encoder,

- Selecting a reference frame for affine combination applied in the decoder and/or encoder,
- Selecting the candidates to be combined applied in the decoder and/or encoder,
- A TV, set-top box, cell phone, tablet, or other electronic device that performs affine motion estimation and/or compensation according to any of the embodiments described,
- A TV, set-top box, cell phone, tablet, or other electronic device that performs affine motion estimation and/or compensation according to any of the embodiments described, and that displays (e.g. using a monitor, screen, or other type of display) a resulting image.
- A TV, set-top box, cell phone, tablet, or other electronic device that tunes (e.g. using a tuner) a channel to receive a signal including an encoded image, and performs affine motion compensation according to any of the embodiments described,
- A TV, set-top box, cell phone, tablet, or other electronic device that receives (e.g. using an antenna) a signal over the air that includes an encoded image, and performs affine motion compensation according to any of the embodiments described.

CLAIMS

1. A method for video encoding, comprising:

obtaining a first set of affine motion model candidates for a current block;

obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and

encoding the current block based on one or more of the affine motion model candidates in the first set and the second set.

2. A method for video decoding, comprising:

obtaining a first set of affine motion model candidates for a current block;

obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and

decoding the current block based on one or more of the affine motion model candidates in the first set and the second set.

3. An apparatus for video encoding, comprising one or more processors, wherein the one or more processors are configured to:

obtain a first set of affine motion model candidates for a current block;

obtain a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and

encode the current block based on based on one or more of the affine motion model candidates in the first set and the second set.

4. An apparatus for video decoding, comprising one or more processors, wherein said one or more processors are configured to:

obtain a first set of affine motion model candidates for a current block;

obtain a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set; and

decode the current block based on one or more of the affine motion model candidates in the first set and the second set.

5. The method of claim 1 or 2, or the apparatus of claim 3 or 4, wherein the combination of one or more of the affine motion model candidates in the first set comprises pair-wise averaging of two of the affine motion model candidates in the first set.

6. The method of any one of claims 1, 2 and 5, or the apparatus of any one of claims 3-5, wherein the first set comprises at least a first list and a second list of affine motion model candidates.

7. The method of any one of claims 1, 2, 5 and 6, or the apparatus of any one of claims 3-6, wherein each of the first set and the second set of affine motion model candidates comprise two or three motion vectors.

8. The method of any one of claims 6 and 7, or the apparatus of any one of claims 6 and 7, wherein the pair-wise averaging is performed using the affine motion model candidates within the first list or within the second list.

9. The method of any one of claims 1, 2 and 5-8, or the apparatus of any one of claims 3-8, wherein selection order of the one or more of the affine motion model candidates in the first set to be combined is predetermined.

10. The method or the apparatus of any one of claims 1, 2 and 5-9, or the apparatus of any one of claims 3-9, wherein the combination is based on a combination of different motion vector components from different affine motion model candidates in the first set.

11. The method of any one of claims 1, 2 and 5-10, or the apparatus of any one of claims 3-10, wherein the combination is based on a combination of the affine motion model candidates in the first set having different reference pictures.

12. The method of any one of claims 10 and 11, or the apparatus of any one of claims 10 and 11, wherein the motion vector components represent control point motion vectors.

13. A bitstream, wherein the bitstream is formed by:

obtaining a first set of affine motion model candidates for a current block;

obtaining a second set of affine motion model candidates using a combination of one or more of the affine motion model candidates in the first set;

encoding the current block based on one or more of the affine motion model candidates in the first set and the second set; and

forming the bitstream comprising the encoded current block.

14. A non-transitory computer readable medium containing data content generated according to the method of any one of claims 1, 2 and 5-12, or the apparatus of any one of claims 3-12.

15. A computer program product comprising instructions for performing the method of any one of claims 1, 2 and 5-12 when executed by one of more processors.

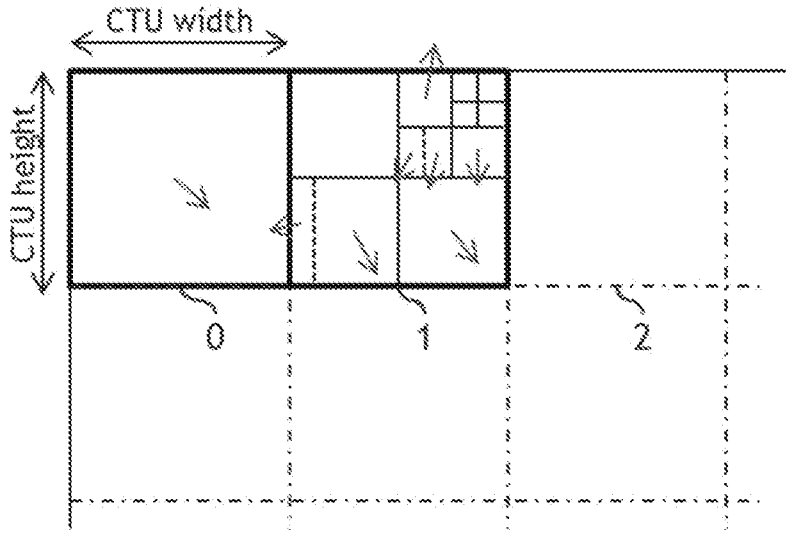

Figure 1

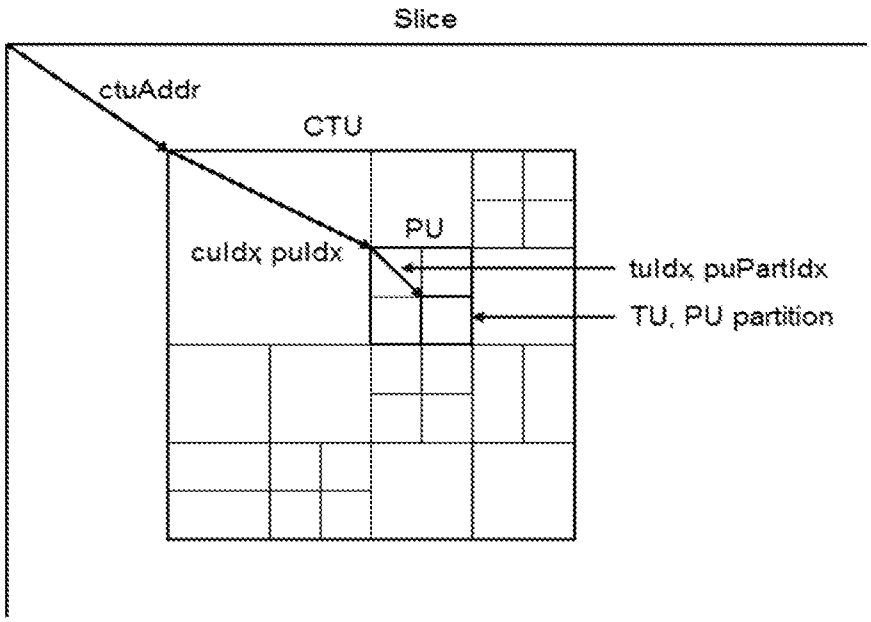

Figure 2

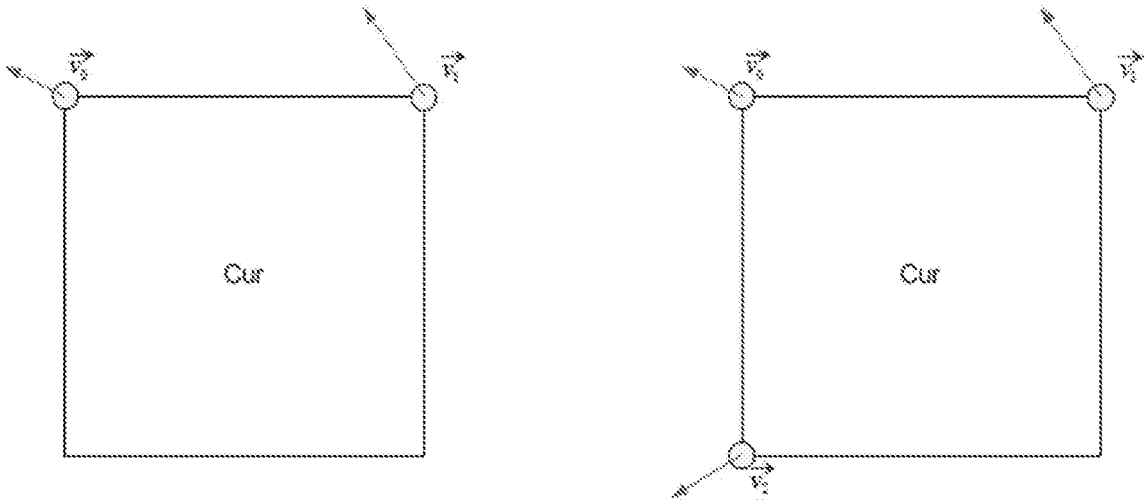

Figure 3

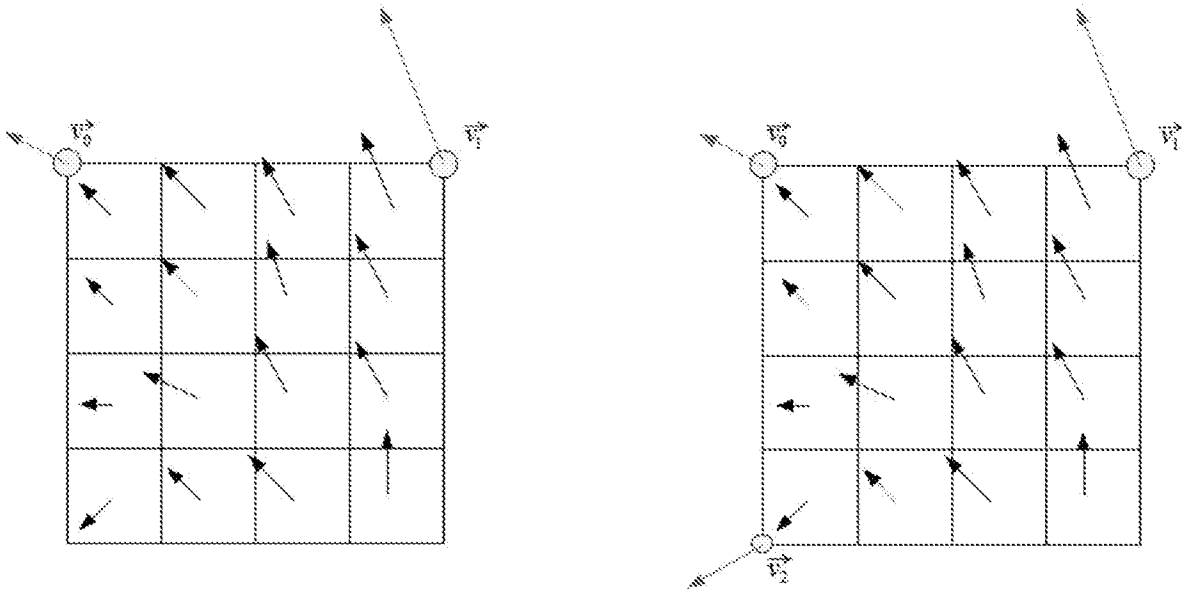

Figure 4

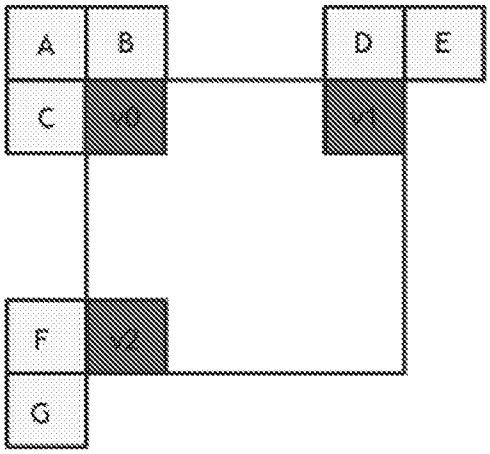

Figure 5

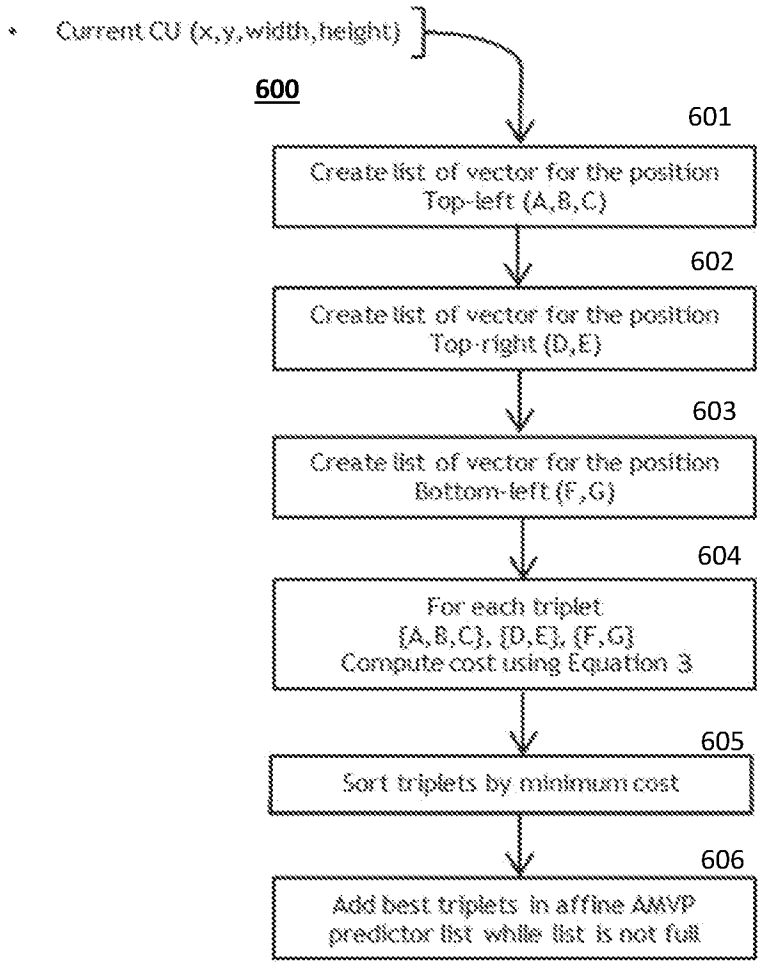

Figure 6

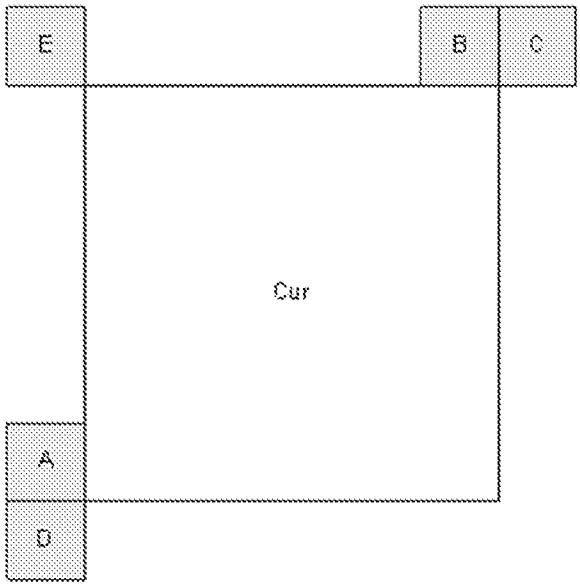

Figure 7

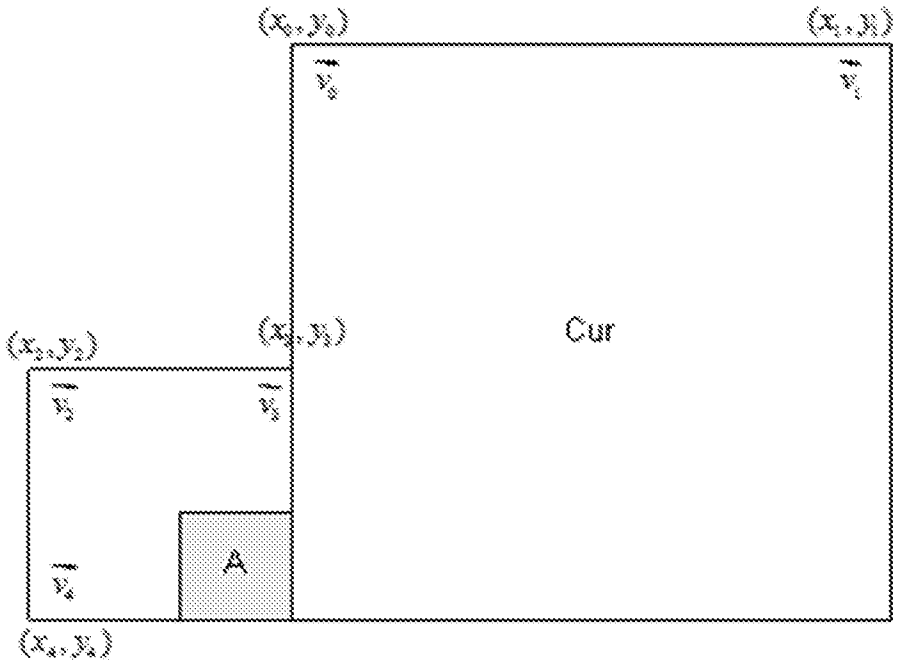

Figure 8

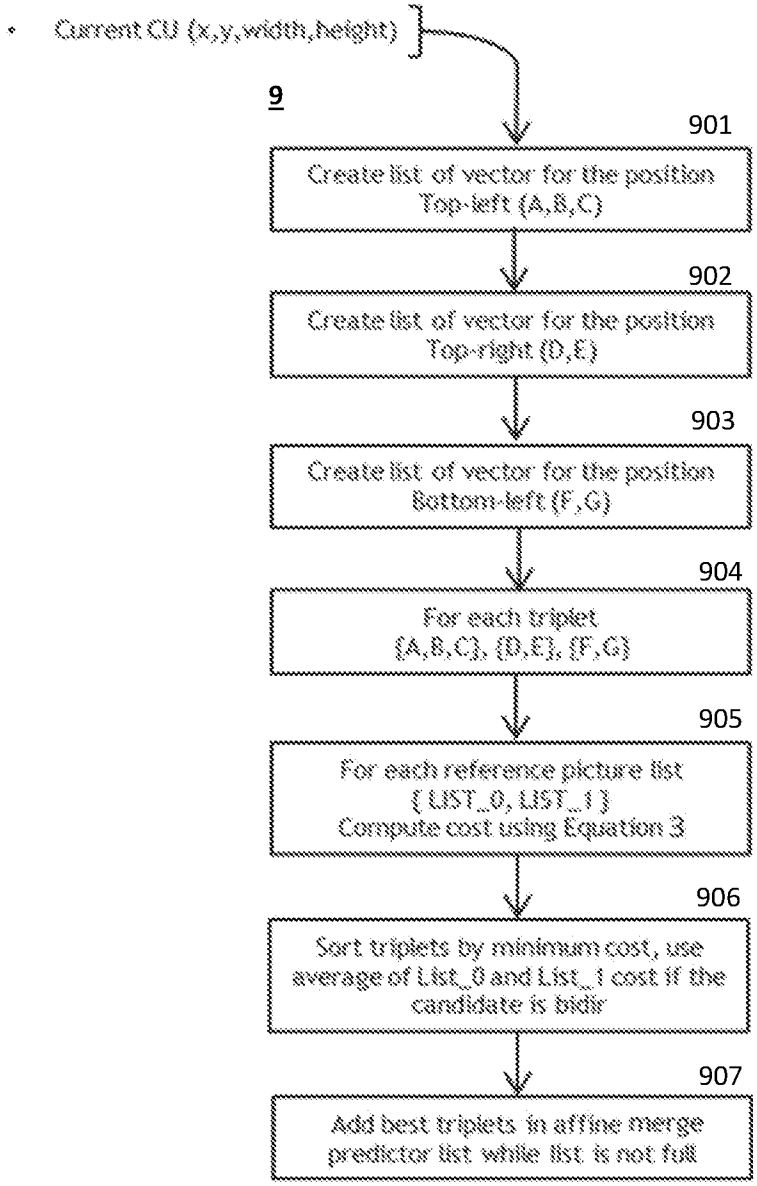
Current CU (x,y,width,height)
9
901
Create list of vector for the position Top-left (A,B,C)
902
Create list of vector for the position Top-right (D,E)
903
Create list of vector for the position Bottom-left (F,G)
904
For each triplet {A,B,C}, {D,E}, {F,G}
905
For each reference picture list { LIST_0, LIST_1 } Compute cost using Equation 3
906
Sort triplets by minimum cost, use average of List_0 and List_1 cost if the candidate is bidir
907
Add best triplets in affine merge predictor list while list is not full

Figure 9

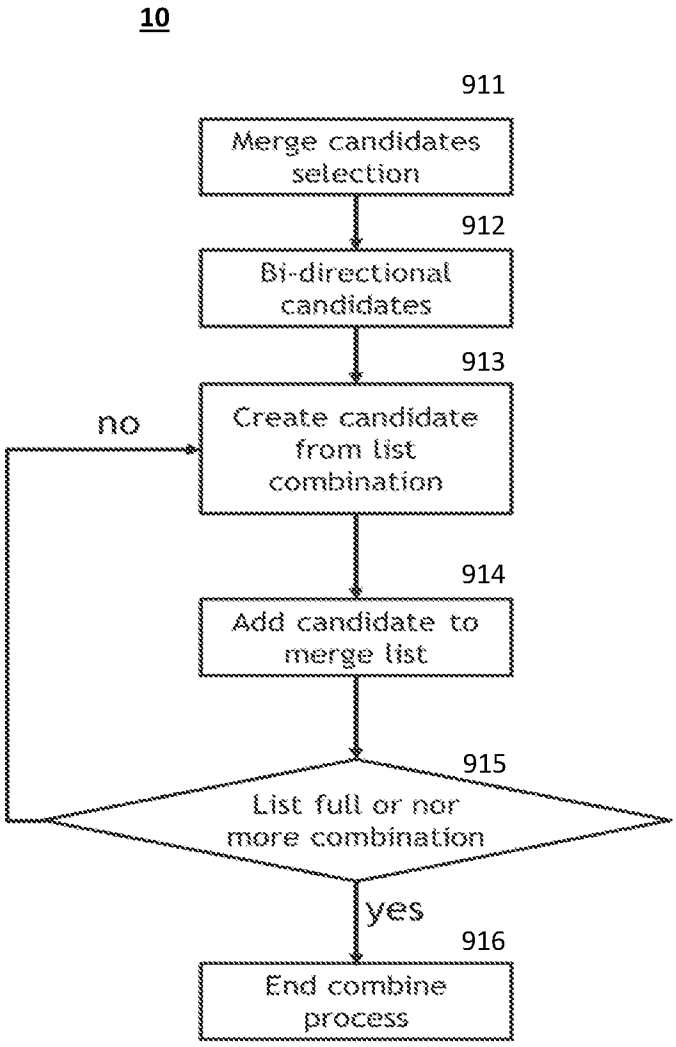
10
911
Merge candidates selection
912
Bi-directional candidates
913
Create candidate from list combination
no
914
Add candidate to merge list
915
List full or nor more combination
yes
916
End combine process

Figure 10

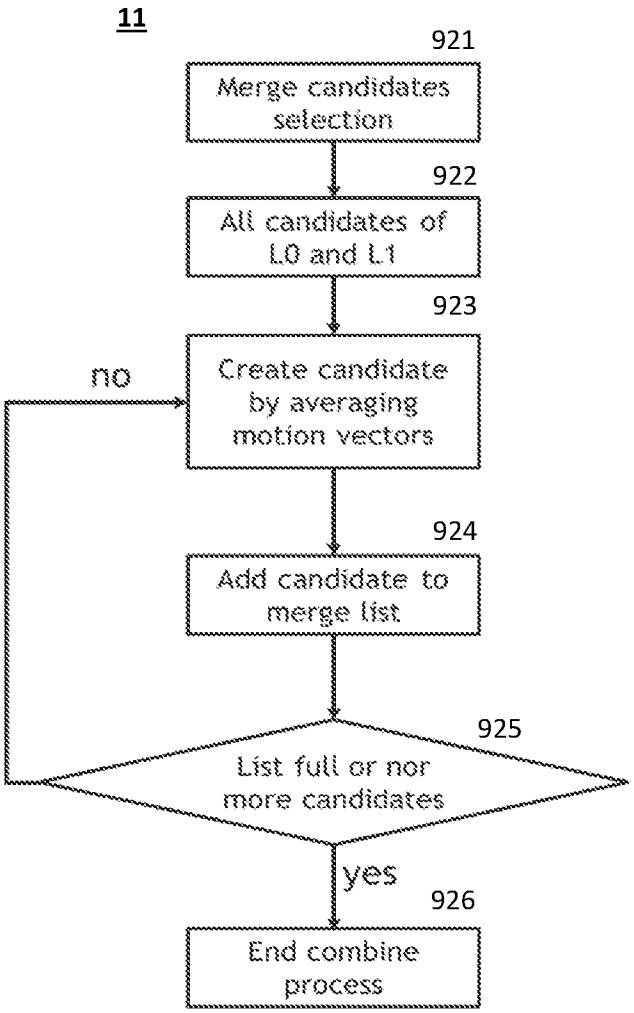
11
921
Merge candidates selection
922
All candidates of L0 and L1
923
Create candidate by averaging motion vectors
no
924
Add candidate to merge list
925
List full or nor more candidates
yes
926
End combine process

Figure 11

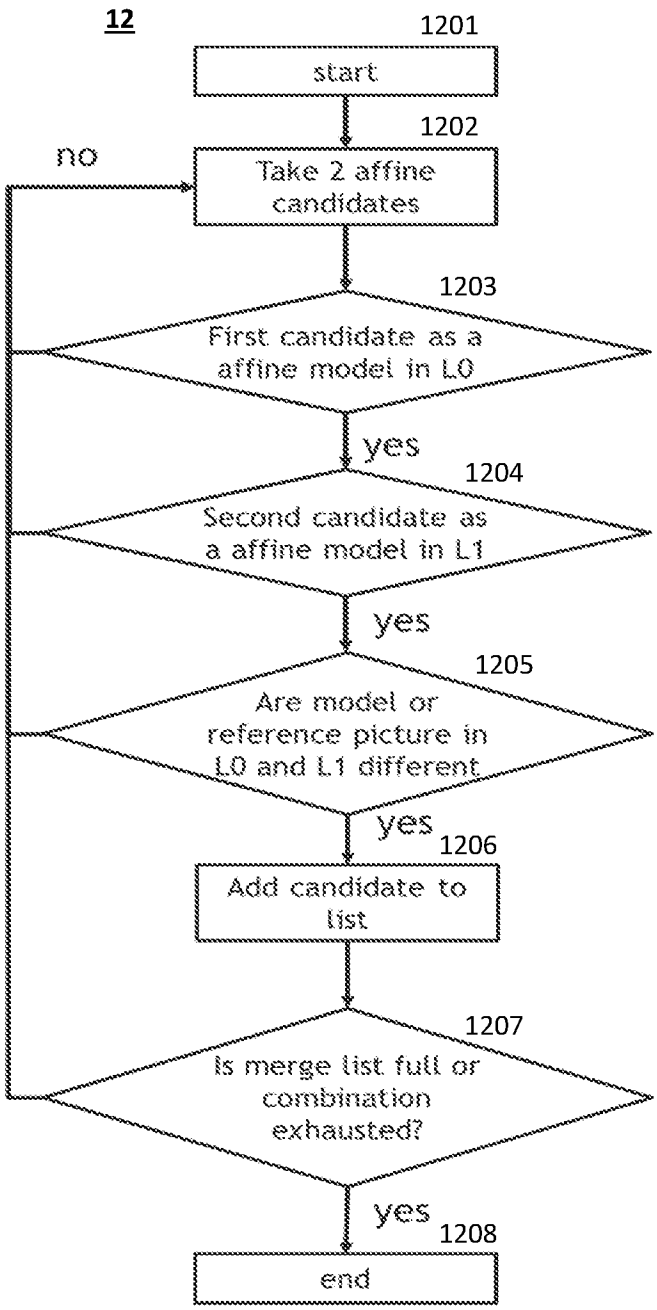
12
1201
start
1202
Take 2 affine candidates
no
1203
First candidate as a affine model in L0
yes
1204
Second candidate as a affine model in L1
yes
1205
Are model or reference picture in L0 and L1 different
yes
1206
Add candidate to list
1207
Is merge list full or combination exhausted?
yes
1208
end

Figure 12

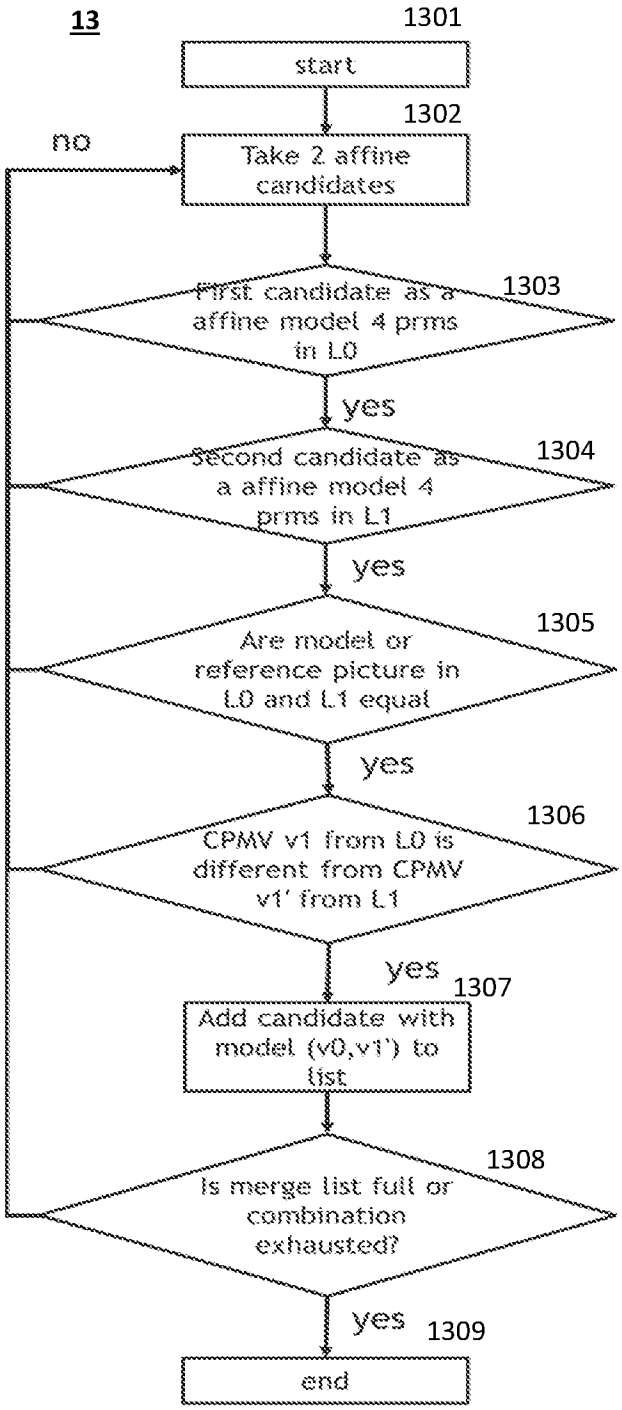
13
1301
start
1302
Take 2 affine candidates
no
First candidate as a affine model 4 prms in L0
1303
yes
Second candidate as a affine model 4 prms in L1
1304
yes
Are model or reference picture in L0 and L1 equal
1305
yes
CPMV v1 from L0 is different from CPMV v1' from L1
1306
yes
Add candidate with model (v0,v1') to list
1307
Is merge list full or combination exhausted?
1308
yes
end
1309

Figure 13

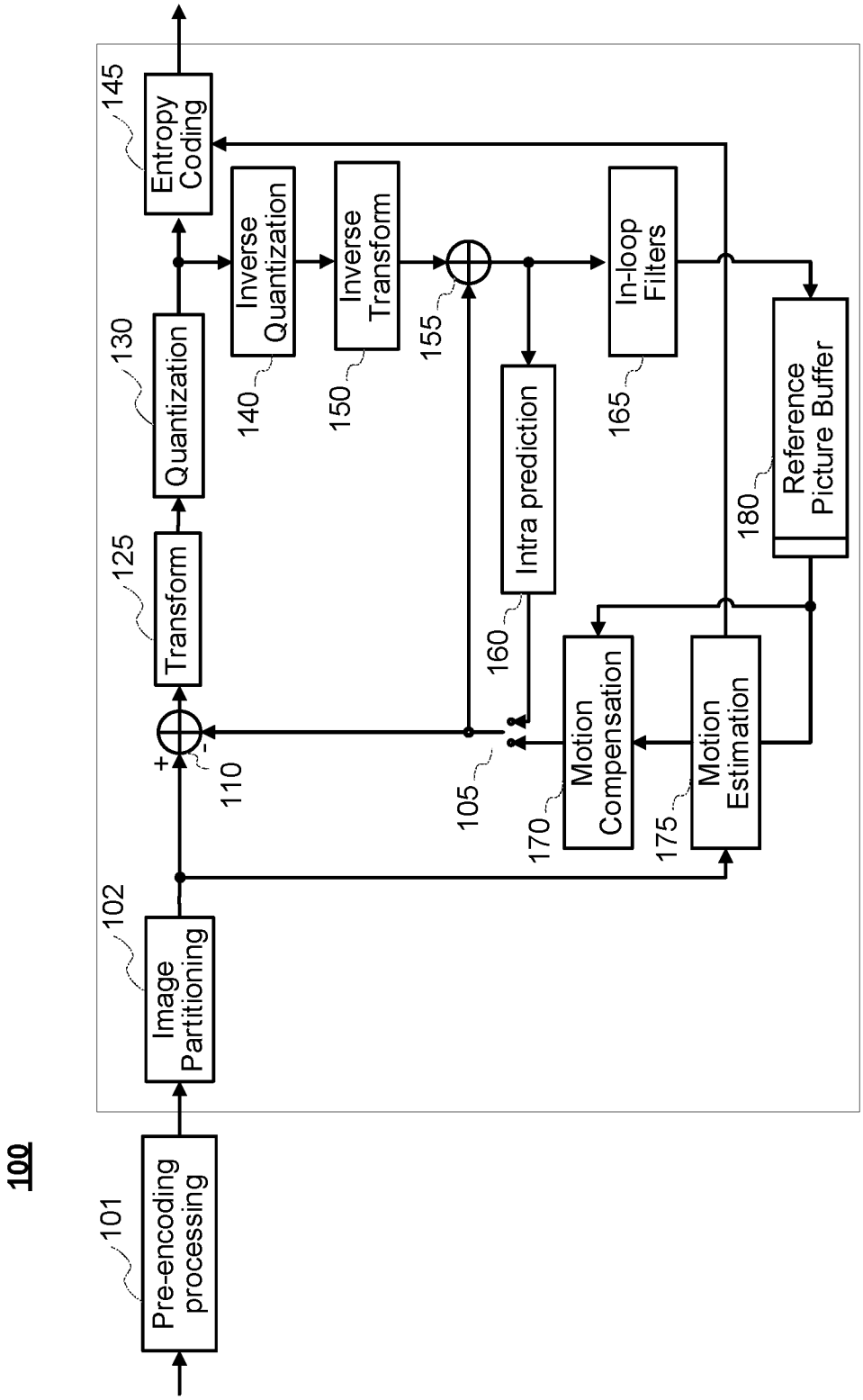
100
101
Pre-encoding processing
102
Image Partitioning
105
110
125
Transform
130
Quantization
140
Inverse Quantization
145
Entropy Coding
150
Inverse Transform
155
160
Intra prediction
165
In-loop Filters
170
Motion Compensation
175
Motion Estimation
180
Reference Picture Buffer

Figure 14

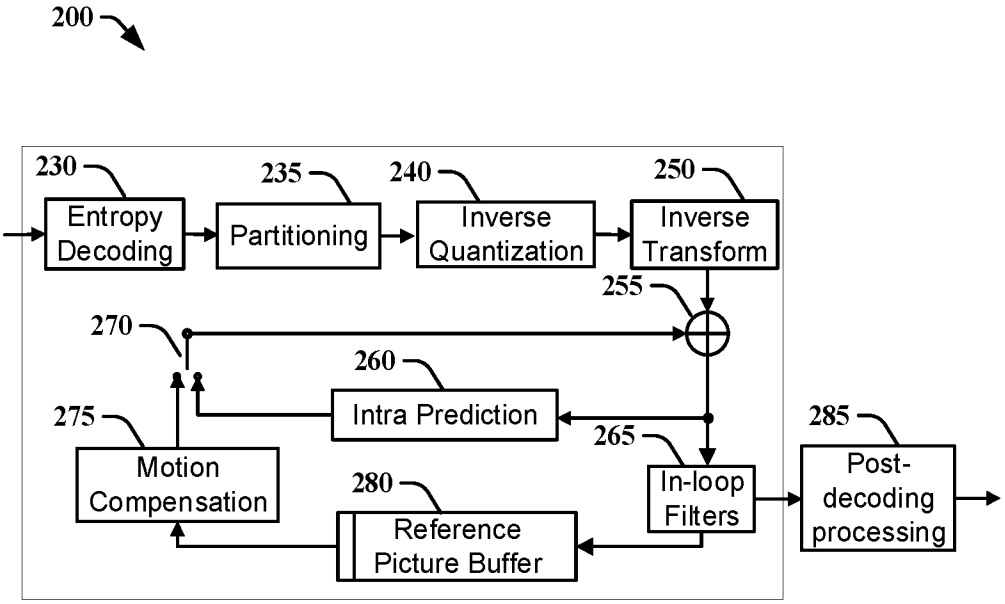

Figure 15

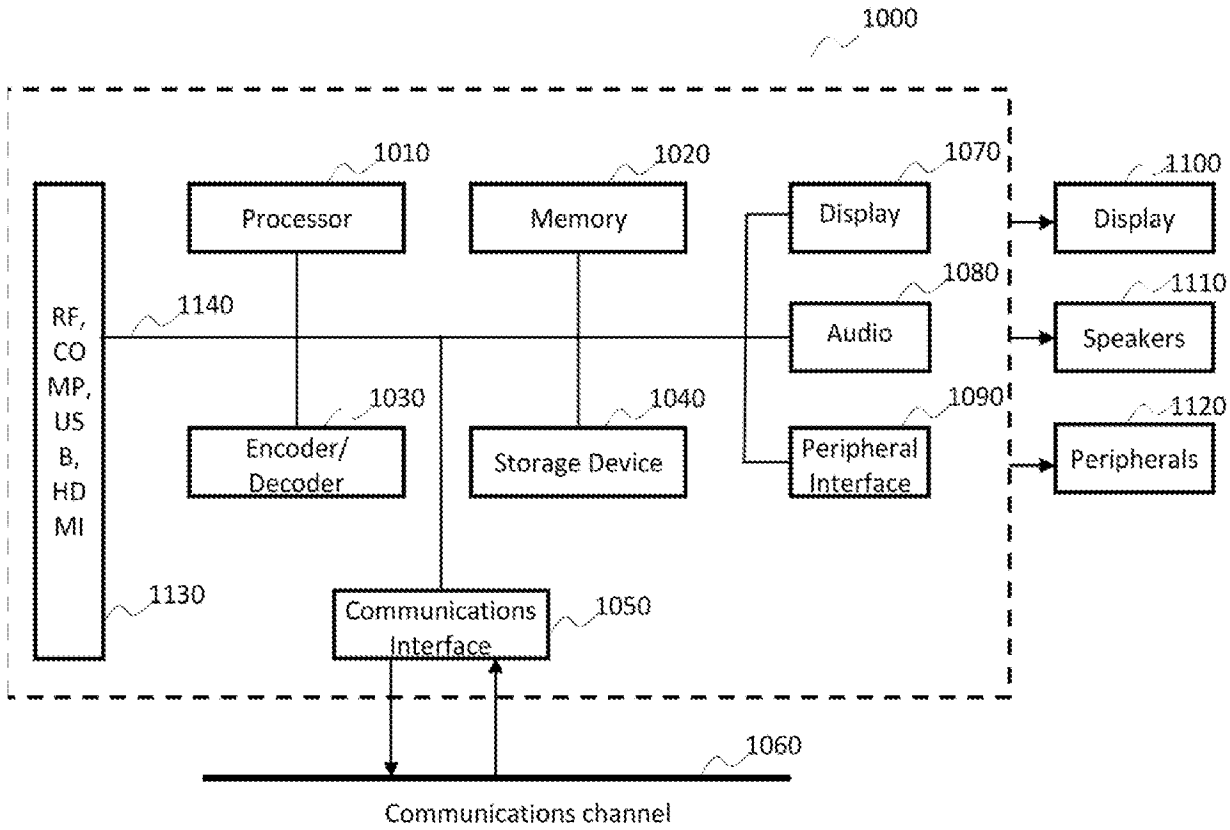

Figure 16

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/062346

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/52 H04N19/513
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, COMPENDEX, 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	WO 2018/128379 A1 (LG ELECTRONICS INC [KR]) 12 July 2018 (2018-07-12) the whole document & US 2019/342547 A1 (LEE JAEHO [KR] ET AL) 7 November 2019 (2019-11-07) paragraph [0207] - paragraph [0219] ----- -/--	1-15

[X] Further documents are listed in the continuation of Box C.

[X] 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	Date of mailing of the international search report
7 February 2020	14/02/2020

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Stoufs, Maryse

2

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/062346

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 (HUAWEI) H ET AL: "CE4: Common base for affine merge mode (Test 4.2.1)", 124. MPEG MEETING; 20181008 - 20181012; MACAO; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. m44392 25 September 2018 (2018-09-25), XP030191096, Retrieved from the Internet: URL:http://phenix.int-evry.fr/mpeg/doc_end_user/documents/124_Macao/wg11/m44392-JVET-L0366-v1-JVET-L0366.zip JVET-L0366.docx [retrieved on 2018-09-25] section 1 "Introduction"	1-15
A	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, table 7	1-15
A	WO 2017/118411 A1 (MEDIATEK INC [CN]) 13 July 2017 (2017-07-13) paragraphs [0096], [0117]	1-15
A	US 2018/324454 A1 (LIN JIAN-LIANG [TW] ET AL) 8 November 2018 (2018-11-08) paragraphs [0046], [0047]	1-15
X,P	EP 3 422 720 A1 (THOMSON LICENSING [FR]) 2 January 2019 (2019-01-02) paragraphs [0093] - [0096]	1-15

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2019/062346

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2018128379 A1	12-07-2018	US 2019342547 A1	07-11-2019
		WO 2018128379 A1	12-07-2018
WO 2017118411 A1	13-07-2017	CN 108432250 A	21-08-2018
		CN 108886619 A	23-11-2018
		GB 2561507 A	17-10-2018
		US 2019028731 A1	24-01-2019
		US 2019158870 A1	23-05-2019
		WO 2017118409 A1	13-07-2017
		WO 2017118411 A1	13-07-2017
US 2018324454 A1	08-11-2018	CN 108353184 A	31-07-2018
		EP 3357245 A1	08-08-2018
		US 2018324454 A1	08-11-2018
		WO 2017076221 A1	11-05-2017
EP 3422720 A1	02-01-2019	NONE	