United States Patent	12395671
Kind Code	B2
Date of Patent	August 19, 2025
Inventor(s)	Zhang; Kai et al.

Interaction between Intra Block Copy mode and inter prediction tools

Abstract

The present disclosure relates to interaction between intra block copy mode and inter prediction tools. A method for video processing includes determining that an intra block copy (IBC) mode is applied to a current video block of a video. In the IBC mode, at least one reference picture used by the current video block is a current picture where the current video block is located. The method also includes making a decision regarding a disabling of a specific coding mode for the current block. The specific coding mode uses a motion vector and a non-current picture to derive a prediction of a video block. The method further includes performing, based on the decision, a conversion between the current video block and the bitstream representation.

Inventors: **Zhang; Kai (San Diego, CA), Zhang; Li (San Diego, CA), Liu; Hongbin (Beijing, CN), Wang; Yue (Beijing, CN)**

Applicant: **Beijing Bytedance Network Technology Co., Ltd.** (Beijing, CN); **Bytedance Inc.** (Los Angeles, CA)

Family ID: **1000008766701**

Assignee: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. (Beijing, CN); BYTEDANCE INC. (Los Angeles, CA)**

Appl. No.: **18/486641**

Filed: **October 13, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20240056599 A1	Feb. 15, 2024

Foreign Application Priority Data

WO	PCT/CN2018/118167	Nov. 29, 2018

Related U.S. Application Data

Publication Classification

Int. Cl.: **H04N19/52** (20140101); **H04N19/159** (20140101); **H04N19/176** (20140101); **H04N19/184** (20140101); **H04N19/51** (20140101); **H04N19/593** (20140101)

U.S. Cl.:

CPC   **H04N19/52** (20141101); **H04N19/159** (20141101); **H04N19/176** (20141101); **H04N19/184** (20141101); **H04N19/51** (20141101); **H04N19/593** (20141101);

Field of Classification Search

CPC: H04N (19/52); H04N (19/184); H04N (19/159); H04N (19/176)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
9591325	12/2016	Li	N/A	N/A
9883197	12/2017	Chen	N/A	N/A
9918105	12/2017	Pang	N/A	N/A
10178403	12/2018	Seregin	N/A	N/A
10264290	12/2018	Xu	N/A	N/A
10484686	12/2018	Xiu	N/A	N/A
10638140	12/2019	Seregin	N/A	N/A
11095917	12/2020	Zhang	N/A	N/A
11115676	12/2020	Zhang	N/A	N/A
2010/0241852	12/2009	Sela	N/A	N/A
2015/0030073	12/2014	Chen	N/A	N/A
2015/0085929	12/2014	Chen	N/A	N/A
2015/0139296	12/2014	Yu	N/A	N/A
2015/0264396	12/2014	Zhang	N/A	N/A
2015/0373359	12/2014	He	N/A	N/A
2016/0100189	12/2015	Pang	N/A	N/A
2016/0241852	12/2015	Gamei	N/A	H04N 19/159
2016/0241858	12/2015	Li	N/A	N/A
2016/0241876	12/2015	Xu	N/A	N/A
2016/0353117	12/2015	Seregin	N/A	N/A
2016/0360234	12/2015	Tourapis	N/A	N/A
2017/0034526	12/2016	Rapaka	N/A	H04N 19/593
2017/0099490	12/2016	Seregin	N/A	N/A
2017/0195677	12/2016	Ye	N/A	N/A

2017/0280159	12/2016	Xu	N/A	N/A
2017/0289566	12/2016	He	N/A	N/A
2017/0332095	12/2016	Zou	N/A	N/A
2018/0070105	12/2017	Jin	N/A	N/A
2018/0109810	12/2017	Xu	N/A	N/A
2018/0109814	12/2017	Chuang	N/A	N/A
2018/0192069	12/2017	Chen	N/A	N/A
2018/0270500	12/2017	Li	N/A	N/A
2018/0288430	12/2017	Chen	N/A	N/A
2018/0376149	12/2017	Zhang	N/A	H04N 19/503
2019/0246128	12/2018	Xu	N/A	H04N 19/159
2020/0036997	12/2019	Li	N/A	N/A
2020/0177910	12/2019	Li	N/A	N/A
2020/0195959	12/2019	Zhang	N/A	N/A
2020/0396465	12/2019	Zhang	N/A	N/A
2021/0160525	12/2020	Zhang	N/A	N/A
2021/0160533	12/2020	Zhang	N/A	N/A

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
105392008	12/2015	CN	N/A
105493505	12/2015	CN	N/A
106416253	12/2016	CN	N/A
106797476	12/2016	CN	N/A
107079161	12/2016	CN	N/A
107409220	12/2016	CN	N/A
107646195	12/2017	CN	N/A
107690809	12/2017	CN	N/A
107852490	12/2017	CN	N/A
108012153	12/2017	CN	N/A
108432250	12/2017	CN	N/A
108702509	12/2017	CN	N/A
108886619	12/2017	CN	N/A
113196772	12/2023	CN	N/A
547928	12/2023	IN	N/A
2017508345	12/2016	JP	N/A
2017522790	12/2016	JP	N/A
2018526881	12/2017	JP	N/A
2016059513	12/2015	KR	N/A
102695787	12/2023	KR	N/A
2015035449	12/2014	WO	N/A
2015192353	12/2014	WO	N/A
2016057938	12/2015	WO	N/A
2016123068	12/2015	WO	N/A
2016123081	12/2015	WO	N/A
2017118409	12/2016	WO	N/A
2017148345	12/2016	WO	N/A
2017157259	12/2016	WO	N/A
2017206804	12/2016	WO	N/A
2018192574	12/2017	WO	N/A

2018200960	12/2017	WO	N/A
2018205954	12/2017	WO	N/A
2018210315	12/2017	WO	N/A

OTHER PUBLICATIONS

Lee et al. "CE4: Simplification of the Common Base for Affine Merge (Test 4.2.2)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and 1SO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macau, CN, Oct. 8-12, 2012, document JVET-L0142, 2018. cited by applicant

Chen et al. "Crosscheck of JVET-L0142 (CE4: Simplification of the Common Base for Affine Merge (Test 4.2.6))," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, document JVET-L0632, 2018. cited by applicant

Huang et al. "CE4.2.5: Simplification of Affine Merge List Construction and Move ATMVP lo Affine Merge List," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, document JVET-L0278, 2018. cited by applicant

Su et al. "CE4-Related: Generalized Bi-Prediction Improvements Combined from JVET-L0197 and JVET-L0296," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, document JVET-L0646, 2018. cited by applicant

Chen et al. "Generalized Bi-Prediction for Inter Coding," Joint Video Exploration Team (JVET) of ITU-T Sg 16 WP 3 and ISO-IEC JTC 1/SC 29/WG 11, 3rd Meeting, Geneva, CH, 26 May to Jun. 1, 2016, document JVET-C0047, 2016. cited by applicant

Su et al. "CE4.4.1: Generalized Bi-Prediction for Intercoding," Joint Video Exploration Team of ISO/IEC JTC 1/SC 29/ WG 11 and ITU-T SG 16, Ljubljana, Jul. 10-18, 2018, document No. JVET-K0248, 2018. cited by applicant

Su et al."CE4-Related: Generalized Bi-Prediction Improvements," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, documentJVET-L0197, 2018. cited by applicant

He et al. "CE4-Related: Encoder Speed-Up and Bug Fix for Generalized Bi-Prediction in BMS-2.1," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, document JVET-L0296, 2018. cited by applicant

Chiang et al. "CE10.1.1: Multi-Hypothesis Prediction for Improving AMVP Mode, Skip or Merge Mode, and Intra Mode," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, document JVET-L0100, 2018. cited by applicant

"High Efficiency Video Coding" Series H: Audiovisual and Multimedia Systems: Infrastructure of Audiovisual Services—Coding of Moving Video, ITU-T, H.265, 2018. cited by applicant

Rosewarne et al. "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 7," Joint Collaborative Team on Video Coding (JCT-VG) ITU-T SG 16 WP3 and 1SO/IEC JTC1/SC29/WG11, 25th Meeting, Chengdu, CN, Oct. 14-21, 2019, document JCTVC-Y1002, 2016. cited by applicant

Chen et al. "Algorithm Description of Joint Exploration Test Model 7 (JEM 7)," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 7th Meeting: Torino, IT, Jul. 13-21, 2017, document JVET-G1001, 2017. cited by applicant

JEM-7.0, Retrieved from the internet: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/ HM-16.6-JEM-7.0, Aug. 16, 2021. cited by applicant

Chen et al. "CE4: Affine Merge Enhancement with Simplification (Test 4.2.2)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting: Macao, CN, Oct. 3-12, 2018, document JVET L0368, 2018. cited by applicant

Li et al. "Non-SCCE1: Unification of Intra BC and Inter Modes," Joint Collaborative Team on Video Coding (JCT-VG) of ITU-T SG 16 WP 3 and 1SO/IEC JTC 1/SC 29/WG 11 18th Meeting, Sapporo, JP, Jun. 30-Jul. 9, 2014, document JCTVC-R0100, 2014. cited by applicant

Chien et al. "Methodology and Reporting Template for Tool Testing," Joint Video Exploration Team {JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 12th Meeting, Macao, CN, Oct. 3-12, 2018, document JVET-L1005, 2018. cited by applicant

Bross et al. "Versatile Video Coding (Draft 3)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, Oct. 3-12, 2018, document JVET-L1001, 2018. cited by applicant

Lin et al. "CE4.2.3: Affine Merge Mode," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, Oct. 3-12, 2018, document JVET-L0088, 2018. cited by applicant

Sullivan et al. "Meeting Report of the 21st Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Warsaw, PL, Jun. 19-26, 2015," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 21st Meeting: Warsaw, PL, Jun. 19-26, 2015, document JCTVC-U1000, 2015. cited by applicant

Document: JVET-J0029-v1, Li, X., et al., "Description of SDR video coding technology proposal by Tencent," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting: San Diego, US, Apr. 10-20, 2018, 34 pages. cited by applicant

Document: JVET-P2002-v1, Chen, J., et al., "Algorithm description for Versatile Video Coding and Test Model 7 (VTM 7)," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 16th Meeting: Geneva, CH, Oct. 1-11, 2019, 90 pages. cited by applicant

Document: JVET-Q0095, Li, G., et al., "CE5-2.2: Multiplication removal for CCALF with coefficient range in [-8, 8]," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting: Brussels, BE, Jan. 7-17, 2020, 4 pages. cited by applicant

Extended European Search Report from European Patent Application No. 19888914.9 dated Nov. 19, 2021 (11 pages). cited by applicant

International Search Report and Written Opinion from International Patent Application No. PCT/CN2019/122183 dated Feb. 28, 2020 (13 pages). cited by applicant

International Search Report and Written Opinion from International Patent Application No. PCT/CN2019/122194 dated Mar. 6, 2020 (10 pages). cited by applicant

International Search Report and Written Opinion from International Patent Application No. PCT/CN2019/122195 dated Mar. 2, 2020 (10 pages). cited by applicant

International Search Report and Written Opinion from International Patent Application No. PCT/CN2019/122198 dated Feb. 28, 2020 (11 pages). cited by applicant

Non-Final Office Action from U.S. Appl. No. 17/167,303 dated Apr. 1, 2021. cited by applicant

Non-Final Office Action from U.S. Appl. No. 17/167,266 dated Mar. 23, 2021. cited by applicant

Office Action from Japanese Patent Application No. 2021-528964 dated Dec. 13, 2022. cited by applicant

Document: JVET-L0198, Wang, S., et al., "Simplification of ATMVP Candidate Derivation," Institute of Digital Media, Peking University, Oct. 5, 2018, 12 pages. cited by applicant

Document: JCTVC-S0172, He, Y., et al., "Non-CE2: Unification of IntraBC mode with inter mode," Joint Video Experts Team on Video Coding (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 19th Meeting: Strasbourg, FR, Oct. 17-24, 2014, 8 pages. cited by applicant

Document: JVET-K1002-v2, Chen, J., et al., "Algorithn description for Versatile Video Coding and Test Model 2 (VTM 2)," Joint Video Experts Team on Video Coding (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting: Ljubljana, SI, Jul. 10-18, 2018, 21 pages. cited by applicant

Extended European Search Report from European Application No. 23213886.7 dated Mar. 6, 2024, 12 pages. cited by applicant

Chinese Notice of Allowance from Chinese Patent Application No. 201980077207.X dated Feb. 24, 2024, 14 pages. With English Translation. cited by applicant

Document: JCTVC-C403-v1, Wiegand, T., et al., "WDI: Working Draft 1 of High-Efficiency Video Coding," Joint Collaborative Team on Video Coding (.TCT-VC) of' ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 3rd Meeting: Guangzhou, CN. Oct. 7-15, 2010, 13 pages. cited by applicant

Japanese Office Action from Japanese Patent Application No. 2023-002737 dated Jul. 2, 2024, 14 pages. cited by applicant

Chinese Notice of Allowance from Chinese Patent Application No. 201980077195.0 dated Sep. 16, 2024, 5 pages. cited by applicant

Chinese Office Action from Chinese Patent Application No. 201980077195.0 dated Jun. 28, 2024, 16 pages. cited by applicant

---

Primary Examiner: Pham; Nam D

Attorney, Agent or Firm: Conley Rose, P.C.

Background/Summary

CROSS REFERENCE TO RELATED APPLICATIONS (1) This application is a continuation of U.S. patent application Ser. No. 17/403,707, filed on Aug. 16, 2021, which is a continuation of U.S. patent application Ser. No. 17/167,266, now U.S. Pat. No. 11,115,676, filed on Feb. 4, 2021, which is a continuation of the International Patent Application No. PCT/CN2019/122183, filed on Nov. 29, 2019, which claims the priority to and benefits of International Patent Application No. PCT/CN2018/118167, filed on Nov. 29, 2018. All the aforementioned patent applications are hereby incorporated by reference in their entireties.

TECHNICAL FIELD

(1) The present disclosure relates to video coding techniques, devices and systems.

BACKGROUND

(2) In spite of the advances in video compression, digital video still 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.

SUMMARY

(3) Devices, systems and methods related to digital video coding, and specifically, to motion vector derivation and signaling for Intra-Block-Copy (IBC) in video and image coding are described. The described methods may be applied to both the existing video coding standards (e.g., High Efficiency Video Coding (HEVC)) and future video coding standards or video codecs.

(4) In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining that an Intra-Block-Copy (IBC) mode is applied to a current video block of a video, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture where the current video block is located in; making a decision regarding a disabling of a specific coding mode for the current block; performing, based on the decision, a conversion between the current video block and the bitstream representation, wherein the specific coding mode uses a motion vector and a non-current picture to derive a prediction of a video block.

(5) In another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining that a current video block is coded using a

specific coding mode; making a decision regarding disabling of an Intra-Block-Copy (IBC) mode for the current block based on the determination, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture where the current block is located in; performing, based on the decision, a conversion between the current block and the bitstream representation, wherein the specific coding mode uses a motion vector and a non-current picture to derive a prediction of the current block.

(6) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining an affine mode of a current video block, wherein whether the current video block inherits an affine mode from a neighbouring block depends on a reference list; performing a conversion between a current video block of a video and a bitstream representation of the video consistent based on the determination.

(7) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes performing a conversion between a current video block of a video and a bitstream representation of the video consistent, wherein during the conversion an Intra-Block-Copy (IBC) mode and a merge with motion vector difference (MMVD) mode are used, and wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in, and in the MMVD mode, a motion vector of a video block is derived based on a merge motion candidate list and is further refined by at least one motion vector difference.

(8) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes performing a conversion between a current video block of a video and a bitstream representation of the video, wherein during the conversion an Intra-Block-Copy (IBC) mode and an inter-intra prediction mode are used, and wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in, and in the inter-intra prediction mode, prediction signal of the current video block is generated at least based on an intra prediction signal and an inter prediction signal.

(9) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining, during a conversion between a current video block and a bitstream representation of the current video block, decoded information of at least one coding method different from an IBC mode, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in; determining, based on the decoded information, whether to skip signaling of an IBC flag for the current video block in a bitstream representation of the current video block, wherein the IBC flag relates with the IB mode; and performing the conversion based on the determining.

(10) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes during a conversion between a current video block and a bitstream representation of the current video block, wherein the current video block comprises a luma component that is coded using a first coding structure tree and a chroma component that is coded using a second coding structure tree different from the first coding structure tree, deriving first motion vector information for a chroma block from second motion vector information of one or more corresponding blocks of the luma component of the current video block; performing the conversion based on the first motion vector information and the second vector information.

(11) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining whether an Intra Block Copy (IBC) mode, in which at least one reference picture used by the current video block is identical to a picture in which the current video block is located , is applied for the current video block; signaling whether the IBC mode is applied for the current video block; refraining from signaling flags for a

group of prediction modes based on determining that the IBC mode is applied for the current video block; and performing the conversion for the current video block based on the application of the IBC mode.

(12) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes parsing a bitstream to determining whether an Intra Block Copy (IBC) mode, in which at least one reference picture used by the current video block is identical to a picture in which the current video block is located, is applied for the current video block; performing the conversion for the current video block based on the application of the IBC mode, wherein flags for a group of prediction modes are excluded from the bitstream based on determining that the current picture referencing (CPR) is applied for the current video block.

(13) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes in a process of acquiring motion information of a current video block, during a conversion between the current video block and a bitstream representation of the current video block, wherein the motion information of the current video block is based on at least one affine model of at least one neighboring block of the current video block; determining whether a neighboring block of the current video block uses an Intra-Block Copy(IBC) mode in which at least one reference picture used by the neighboring video block is identical to a picture in which the neighboring block is located; disabling, for the current video block, to derive an affine candidate from the neighboring block based on determining that the neighbouring block uses the IBC mode; and performing the conversion based on the motion information of the current video block.

(14) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes splitting a current video block into sub-blocks, during a conversion between a current video block and a bitstream representation of the current video block based on subblock-based temporal motion vector prediction (SbTMVP) mode; identifying, based on motion information of a spatial neighboring block of the current video block, a corresponding block, for a sub-block, in a motion source picture with a temporal vector; generating, based on coded mode information of the corresponding block, a motion vector and a reference index of the sub-block from the corresponding block of the sub-block; performing the conversion based on the generated motion vector and reference index.

(15) In yet another representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes splitting a current video block into sub-blocks, during a conversion between a current video block and a bitstream representation of the current video block based on subblock-based temporal motion vector prediction (SbTMVP) mode; identifying, based on coded mode information of a spatial neighboring block of the current video block, a corresponding block, for a sub-block, in a motion source picture with a temporal vector; generating a motion vector and a reference index of the sub-block from the corresponding block of the sub-block; performing the conversion based on the generated motion vector and reference index.

(16) In yet another representative aspect, the above-described method is embodied in the form of processor-executable code and stored in a computer-readable program medium.

(17) 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.

(18) In yet another representative aspect, a video decoder apparatus may implement a method as described herein.

(19) The above and other aspects and features of the disclosed technology are described in greater detail in the drawings, the description and the claims.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
(1) FIG. **1** shows an example of constructing a merge candidate list.
(2) FIG. **2** shows an example of positions of spatial candidates.
(3) FIG. **3** shows an example of candidate pairs subject to a redundancy check of spatial merge candidates.
(4) FIGS. **4**A and **4**B show examples of the position of a second prediction unit (PU) based on the size and shape of the current block.
(5) FIG. **5** shows an example of motion vector scaling for temporal merge candidates.
(6) FIG. **6** shows an example of candidate positions for temporal merge candidates.
(7) FIG. **7** shows an example of generating a combined bi-predictive merge candidate.
(8) FIG. **8** shows an example of constructing motion vector prediction candidates.
(9) FIG. **9** shows an example of motion vector scaling for spatial motion vector candidates.
(10) FIG. **10** shows an example of motion prediction using the alternative temporal motion vector prediction (ATMVP) algorithm for a coding unit (CU).
(11) FIG. **11** shows an example of a coding unit (CU) with sub-blocks and neighboring blocks used by the spatial-temporal motion vector prediction (STMVP) algorithm.
(12) FIG. **12** shows an example flowchart for encoding with different motion vector (MV) precisions.
(13) FIGS. **13**A and **13**B show examples of the simplified 4-parameter affine model and the simplified 6-parameter affine model, respectively.
(14) FIG. **14** shows an example of an affine motion vector field (MVF) per sub-block.
(15) FIGS. **15**A and **15**B show examples of the 4-parameter and 6-parameter affine models, respectively.
(16) FIG. **16** shows an example of motion vector prediction for AF_INTER for inherited affine candidates.
(17) FIG. **17** shows an example of motion vector prediction for AF_INTER for constructed affine candidates.
(18) FIGS. **18**A and **18**B show example candidates for the AF_MERGE affine motion mode.
(19) FIG. **19** shows an example of candidate positions for affine merge mode.
(20) FIG. **20** shows an example of intra-picture block copy.
(21) FIG. **21** shows an example of a valid corresponding region in a collocated picture.
(22) FIG. **22** shows an example of the coding flow for history-based motion vector prediction (HMVP) candidates.
(23) FIG. **23** shows an example of a modified merge list construction process.
(24) FIG. **24** shows a flowchart of an example method for video processing.
(25) FIG. **25** shows a flowchart of another example method for video processing.
(26) FIG. **26** shows a flowchart of yet another example method for video processing.
(27) FIG. **27** shows a flowchart of yet another example method for video processing.
(28) FIG. **28** is a block diagram of an example of a hardware platform for implementing a visual media decoding or a visual media encoding technique described in the present disclosure.
(29) FIG. **29** shows a flowchart of yet another example method for video processing.
(30) FIG. **30** shows a flowchart of yet another example method for video processing.
(31) FIG. **31** shows a flowchart of yet another example method for video processing.
(32) FIG. **32** shows a flowchart of yet another example method for video processing.
(33) FIG. **33** shows a flowchart of yet another example method for video processing.
(34) FIG. **34** shows a flowchart of yet another example method for video processing.
(35) FIG. **35** shows a flowchart of yet another example method for video processing.

(36) FIG. **36** shows a flowchart of yet another example method for video processing.

(37) FIG. **37** shows a flowchart of yet another example method for video processing.

(38) FIG. **38** shows a flowchart of yet another example method for video processing.

(39) FIG. **39** shows a flowchart of yet another example method for video processing.

(40) FIG. **40** shows a flowchart of yet another example method for video processing.

DETAILED DESCRIPTION

(41) Due to the increasing demand of 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.

(42) Embodiments of the disclosed technology may be applied to existing video coding standards (e.g., HEVC, H.265) and future standards to improve compression performance. Section headings are used in the present disclosure 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.

(43) In the present disclosure, the term "video processing" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa.

1 Examples of Inter-Prediction in HEVC/H.265

(44) Video coding standards have significantly improved over the years, and now provide, in part, high coding efficiency and support for higher resolutions. Recent standards such as HEVC and H.265 are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized.

1.1 Examples of Prediction Modes

(45) Each inter-predicted PU (prediction unit) has motion parameters for one or two reference picture lists. In some embodiments, motion parameters include a motion vector and a reference picture index. In other embodiments, the usage of one of the two reference picture lists may also be signaled using inter_pred_idc. In yet other embodiments, motion vectors may be explicitly coded as deltas relative to predictors.

(46) When a CU is coded with skip mode, one PU is associated with the CU, and there are no significant residual coefficients, no coded motion vector delta or reference picture index. A merge mode is specified whereby the motion parameters for the current PU are obtained from neighboring PUs, including spatial and temporal candidates. The merge mode can be applied to any inter-predicted PU, not only for skip mode. The alternative to merge mode is the explicit transmission of motion parameters, where motion vector, corresponding reference picture index for each reference picture list and reference picture list usage are signaled explicitly per each PU.

(47) When signaling indicates that one of the two reference picture lists is to be used, the PU is produced from one block of samples. This is referred to as 'uni-prediction'. Uni-prediction is available both for P-slices and B-slices.

(48) When signaling indicates that both of the reference picture lists are to be used, the PU is produced from two blocks of samples. This is referred to as 'bi-prediction'. Bi-prediction is available for B-slices only.

(49) Reference Picture List

(50) In HEVC, the term inter prediction is used to denote prediction derived from data elements (e.g., sample values or motion vectors) of reference pictures other than the current decoded picture. Like in H.264/AVC, a picture can be predicted from multiple reference pictures. The reference pictures that are used for inter prediction are organized in one or more reference picture lists. The reference picture index identifies which of the reference pictures in the list should be used for creating the prediction signal.

(51) A single reference picture list, List 0, is used for a P slice and two reference picture lists, List 0 and List 1 are used for B slices. It should be noted reference pictures included in List 0/1 could be from past and future pictures in terms of capturing/display order.

1.1.1 Embodiments of Constructing Candidates for Merge Mode

(52) When a PU is predicted using merge mode, an index pointing to an entry in the merge candidates list is parsed from the bitstream and used to retrieve the motion information. The construction of this list can be summarized according to the following sequence of steps: Step 1: Initial candidates derivation Step 1.1: Spatial candidates derivation Step 1.2: Redundancy check for spatial candidates Step 1.3: Temporal candidates derivation Step 2: Additional candidates insertion Step 2.1: Creation of bi-predictive candidates Step 2.2: Insertion of zero motion candidates

(53) FIG. **1** shows an example of constructing a merge candidate list based on the sequence of steps summarized above. For spatial merge candidate derivation, a maximum of four merge candidates are selected among candidates that are located in five different positions. For temporal merge candidate derivation, a maximum of one merge candidate is selected among two candidates. Since constant number of candidates for each PU is assumed at decoder, additional candidates are generated when the number of candidates does not reach to maximum number of merge candidate (MaxNumMergeCand) which is signaled in slice header. Since the number of candidates is constant, index of best merge candidate is encoded using truncated unary binarization (TU). If the size of CU is equal to 8, all the PUs of the current CU share a single merge candidate list, which is identical to the merge candidate list of the 2N×2N prediction unit.

1.1.2 Constructing Spatial Merge Candidates

(54) In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in FIG. **2**. The order of derivation is A.sub.1, B.sub.1, B.sub.0, A.sub.0 and B.sub.2. Position B.sub.2 is considered only when any PU of position A.sub.1, B.sub.1, B.sub.0, A.sub.0 is not available (e.g., because it belongs to another slice or tile) or is intra coded. After candidate at position A.sub.1 is added, the addition of the remaining candidates is subject to a redundancy check which ensures that candidates with same motion information are excluded from the list so that coding efficiency is improved.

(55) To reduce computational complexity, not all possible candidate pairs are considered in the mentioned redundancy check. Instead only the pairs linked with an arrow in FIG. **3** are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information. Another source of duplicate motion information is the "second PU" associated with partitions different from 2N×2N. As an example, FIG. **4**A and **4**B depict the second PU for the case of N×2N and 2N×N, respectively. When the current PU is partitioned as N×2N, candidate at position A.sub.1 is not considered for list construction. In some embodiments, adding this candidate may lead to two prediction units having the same motion information, which is redundant to just have one PU in a coding unit. Similarly, position B.sub.1 is not considered when the current PU is partitioned as 2N×N.

1.1.3 Constructing Temporal Merge Candidates

(56) In this step, only one candidate is added to the list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on co-located PU belonging to the picture which has the smallest picture order count (POC) difference with current picture within the given reference picture list. The reference picture list to be used for derivation of the co-located PU is explicitly signaled in the slice header.

(57) FIG. **5** shows an example of the derivation of the scaled motion vector for a temporal merge candidate (as the dotted line), which is scaled from the motion vector of the co-located PU using the POC distances, tb and td, where tb is defined to be the POC difference between the reference picture of the current picture and the current picture and td is defined to be the POC difference between the reference picture of the co-located picture and the co-located picture. The reference picture index of temporal merge candidate is set equal to zero. For a B-slice, two motion vectors, one is for reference picture list 0 and the other is for reference picture list 1, are obtained and combined to make the bi-predictive merge candidate.

(58) In the co-located PU (Y) belonging to the reference frame, the position for the temporal candidate is selected between candidates C.sub.0 and C.sub.1, as depicted in FIG. **6**. If PU at position C.sub.0 is not available, is intra coded, or is outside of the current CTU, position C.sub.1 is used. Otherwise, position C.sub.0 is used in the derivation of the temporal merge candidate.

1.1.4 Constructing Additional Types of Merge Candidates

(59) Besides spatio-temporal merge candidates, there are two additional types of merge candidates: combined bi-predictive merge candidate and zero merge candidate. Combined bi-predictive merge candidates are generated by utilizing spatio-temporal merge candidates. Combined bi-predictive merge candidate is used for B-Slice only. The combined bi-predictive candidates are generated by combining the first reference picture list motion parameters of an initial candidate with the second reference picture list motion parameters of another. If these two tuples provide different motion hypotheses, they will form a new bi-predictive candidate.

(60) FIG. **7** shows an example of this process, wherein two candidates in the original list (710, on the left), which have mvL0 and refId×L0 or mvL1 and refId×L1, are used to create a combined bi-predictive merge candidate added to the final list (720, on the right).

(61) Zero motion candidates are inserted to fill the remaining entries in the merge candidates list and therefore hit the MaxNumMergeCand capacity. These candidates have zero spatial displacement and a reference picture index which starts from zero and increases every time a new zero motion candidate is added to the list. The number of reference frames used by these candidates is one and two for uni- and bi-directional prediction, respectively. In some embodiments, no redundancy check is performed on these candidates.

(62) In some embodiments, the following steps are performed in order till the merge list is full: (1) Set variable numRef to either number of reference picture associated with list 0 for P slice, or the minimum number of reference pictures in two lists for B slice; (2) Add non-repeated zero motion candidates:

(63) For variable i being 0 . . . numRef-1, add a default motion candidate with MV set to (0, 0) and reference picture index set to i for list 0 (if P slice), or for both lists (if B slice). (3) Add repeated zero motion candidates with MV set to (0, 0), reference picture index of list 0 set to 0 (if P slice) and reference picture indices of both lists set to 0 (if B slice).

1.2 Embodiments of Advanced Motion Vector Prediction (AMVP)

(64) AMVP exploits spatio-temporal correlation of motion vector with neighboring PUs, which is used for explicit transmission of motion parameters. It constructs a motion vector candidate list by firstly checking availability of left, above temporally neighboring PU positions, removing redundant candidates and adding zero vector to make the candidate list to be constant length. Then, the encoder can select the best predictor from the candidate list and transmit the corresponding index indicating the chosen candidate. Similarly with merge index signaling, the index of the best motion vector candidate is encoded using truncated unary. The maximum value to be encoded in this case is 2 (see FIG. **8**). In the following sections, details about derivation process of motion vector prediction candidate are provided.

1.2.1 Examples of Deriving AMVP Candidates

(65) FIG. **8** summarizes derivation process for motion vector prediction candidate, and may be implemented for each reference picture list with refidx as an input.

(66) In motion vector prediction, two types of motion vector candidates are considered: spatial motion vector candidate and temporal motion vector candidate. For spatial motion vector candidate derivation, two motion vector candidates are eventually derived based on motion vectors of each PU located in five different positions as previously shown in FIG. **2**.

(67) For temporal motion vector candidate derivation, one motion vector candidate is selected from two candidates, which are derived based on two different co-located positions. After the first list of spatio-temporal candidates is made, duplicated motion vector candidates in the list are removed. If the number of potential candidates is larger than two, motion vector candidates whose reference picture index within the associated reference picture list is larger than 1 are removed from the list. If the number of spatio-temporal motion vector candidates is smaller than two, additional zero motion vector candidates is added to the list.

1.2.2 Constructing Spatial Motion Vector Candidates

(68) In the derivation of spatial motion vector candidates, a maximum of two candidates are considered among five potential candidates, which are derived from PUs located in positions as previously shown in FIG. **2**, those positions being the same as those of motion merge. The order of derivation for the left side of the current PU is defined as A.sub.0, A.sub.1, and scaled A.sub.0, scaled A.sub.1. The order of derivation for the above side of the current PU is defined as B.sub.0, B.sub.1, B.sub.2, scaled B.sub.0, scaled B.sub.1, scaled B.sub.2. For each side there are therefore four cases that can be used as motion vector candidate, with two cases not required to use spatial scaling, and two cases where spatial scaling is used. The four different cases are summarized as follows: No spatial scaling (1) Same reference picture list, and same reference picture index (same POC) (2) Different reference picture list, but same reference picture (same POC) Spatial scaling (3) Same reference picture list, but different reference picture (different POC) (4) Different reference picture list, and different reference picture (different POC)

(69) The no-spatial-scaling cases are checked first followed by the cases that allow spatial scaling. Spatial scaling is considered when the POC is different between the reference picture of the neighboring PU and that of the current PU regardless of reference picture list. If all PUs of left candidates are not available or are intra coded, scaling for the above motion vector is allowed to help parallel derivation of left and above MV candidates. Otherwise, spatial scaling is not allowed for the above motion vector.

(70) As shown in the example in FIG. **9**, for the spatial scaling case, the motion vector of the neighboring PU is scaled in a similar manner as for temporal scaling. One difference is that the reference picture list and index of current PU is given as input; the actual scaling process is the same as that of temporal scaling.

1.2.3 Constructing Temporal Motion Vector Candidates

(71) Apart from the reference picture index derivation, all processes for the derivation of temporal merge candidates are the same as for the derivation of spatial motion vector candidates (as shown in the example in FIG. **6**). In some embodiments, the reference picture index is signaled to the decoder.

2 Example of Inter Prediction Methods in Joint Exploration Model (JEM)

(72) 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).

2.1 Examples of Sub-CU Based Motion Vector Prediction

(73) In the JEM with quadtrees plus binary trees (QTBT), each CU can have at most one set of motion parameters for each prediction direction. In some embodiments, two sub-CU level motion

vector prediction methods are considered in the encoder by splitting a large CU into sub-CUs and deriving motion information for all the sub-CUs of the large CU. Alternative temporal motion vector prediction (ATMVP) method allows each CU to fetch multiple sets of motion information from multiple blocks smaller than the current CU in the collocated reference picture. In spatial-temporal motion vector prediction (STMVP) method motion vectors of the sub-CUs are derived recursively by using the temporal motion vector predictor and spatial neighbouring motion vector. In some embodiments, and to preserve more accurate motion field for sub-CU motion prediction, the motion compression for the reference frames may be disabled.

2.1.1 Examples of Alternative Temporal Motion Vector Prediction (ATMVP)

(74) 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.

(75) FIG. **10** shows an example of ATMVP motion prediction process for a CU **1000**. The ATMVP method predicts the motion vectors of the sub-CUs **1001** within a CU **1000** in two steps. The first step is to identify the corresponding block **1051** in a reference picture **1050** with a temporal vector. The reference picture **1050** is also referred to as the motion source picture. The second step is to split the current CU **1000** into sub-CUs **1001** and obtain the motion vectors as well as the reference indices of each sub-CU from the block corresponding to each sub-CU.

(76) In the first step, a reference picture **1050** and the corresponding block is determined by the motion information of the spatial neighboring blocks of the current CU **1000**. To avoid the repetitive scanning process of neighboring blocks, the first merge candidate in the merge candidate list of the current CU **1000** is used. The first available motion vector as well as its associated reference picture 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.

(77) In the second step, a corresponding block of the sub-CU **1051** is identified by the temporal vector in the motion source picture **1050**, 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×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 MVx (e.g., the motion vector corresponding to reference picture list X) to predict motion vector MVy (e.g., with X being equal to 0 or 1 and Y being equal to 1−X) for each sub-CU.

2.1.2 Examples of Spatial-Temporal Motion Vector Prediction (STMVP)

(78) In the STMVP method, the motion vectors of the sub-CUs are derived recursively, following raster scan order. FIG. **11** shows an example of one CU with four sub-blocks and neighboring blocks. Consider an 8×8 CU **1100** that includes four 4×4 sub-CUs A (**1101**), B (**1102**), C (**1103**), and D (**1104**). The neighboring 4×4 blocks in the current frame are labelled as a (**1111**), b (**1112**), c (**1113**), and d (**1114**).

(79) The motion derivation for sub-CU A starts by identifying its two spatial neighbors. The first neighbor is the N×N block above sub-CU A **1101** (block c **1113**). If this block c (**1113**) is not available or is intra coded the other N×N blocks above sub-CU A (**1101**) are checked (from left to right, starting at block c **1113**). The second neighbor is a block to the left of the sub-CU A **1101** (block b **1112**). If block b (**1112**) is not available or is intra coded other blocks to the left of sub-CU A **1101** are checked (from top to bottom, staring at block b **1112**). 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 **1101** is derived by following the same procedure of TMVP derivation as specified in HEVC. The motion information of the collocated block at block D **1104** 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.

2.1.3 Examples of Sub-CU Motion Prediction Mode Signaling

(80) In some embodiments, the sub-CU modes are enabled as additional merge candidates and there is no additional syntax element required to signal the modes. Two additional merge candidates are added to merge candidates list of each CU to represent the ATMVP mode and STMVP mode. In other embodiments, up to seven merge candidates may be used, if the sequence parameter set indicates that ATMVP and STMVP are enabled. The encoding logic of the additional merge candidates is the same as for the merge candidates in the HM, which means, for each CU in P or B slice, two more rate distortion (RD) checks may be needed for the two additional merge candidates. In some embodiments, e.g., JEM, all bins of the merge index are context coded by CABAC (Context-based Adaptive Binary Arithmetic Coding). In other embodiments, e.g., HEVC, only the first bin is context coded and the remaining bins are context by-pass coded.

2.2 Example of Inter Prediction Methods in VVC

(81) There are several new coding tools for inter prediction improvement, such as Adaptive motion vector difference resolution (AMVR) for signaling MVD, affine prediction mode, Triangular prediction mode (TPM), ATMVP, Generalized Bi-Prediction (GBi), Bi-directional Optical flow (BIO).

2.2.1 Examples of Coding Block Structures in VVC

(82) In versatile video coding (VVC), a QuadTree/BinaryTree/MulitpleTree (QT/BT/TT) structure is adopted to divide a picture into square or rectangle blocks. Besides QT/BT/TT, separate tree (a.k.a., Dual coding tree) is also adopted in VVC for I-frames. With separate trees, the coding block structure are signaled separately for the luma and chroma components.

2.2.2 Examples of Adaptive Motion Vector Difference Resolution

(83) In some embodiments, motion vector differences (MVDs) (between the motion vector and predicted motion vector of a PU) are signalled in units of quarter luma samples when use_integer_mv_flag is equal to 0 in the slice header. In the JEM, a locally adaptive motion vector resolution (LAMVR) is introduced. In the JEM, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples. The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signalled for each CU that has at least one non-zero MVD components.

(84) For a CU that has at least one non-zero MVD components, a first flag is signalled to indicate whether quarter luma sample MV precision is used in the CU. When the first flag (equal to 1) indicates that quarter luma sample MV precision is not used, another flag is signalled to indicate whether integer luma sample MV precision or four luma sample MV precision is used.

(85) When the first MVD resolution flag of a CU is zero, or not coded for a CU (meaning all MVDs in the CU are zero), the quarter luma sample MV resolution is used for the CU. When a CU uses integer-luma sample MV precision or four-luma-sample MV precision, the MVPs in the AMVP candidate list for the CU are rounded to the corresponding precision.

(86) In the encoder, CU-level RD checks are used to determine which MVD resolution is to be used for a CU. That is, the CU-level RD check is performed three times for each MVD resolution. To accelerate encoder speed, the following encoding schemes are applied in the JEM: During RD check of a CU with normal quarter luma sample MVD resolution, the motion information of the current CU (integer luma sample accuracy) is stored. The stored motion information (after rounding) is used as the starting point for further small range motion vector refinement during the RD check for the same CU with integer luma sample and **4** luma sample MVD resolution so that the time-consuming motion estimation process is not duplicated three times. RD check of a CU

with 4 luma sample MVD resolution is conditionally invoked. For a CU, when RD cost integer luma sample MVD resolution is much larger than that of quarter luma sample MVD resolution, the RD check of 4 luma sample MVD resolution for the CU is skipped.

(87) The encoding process is shown in FIG. **12**. First, ¼ pel MV is tested and the RD cost is calculated and denoted as RDCost0, then integer MV is tested and the RD cost is denoted as RDCost1 . If RDCost1<th*RDCost0 (wherein th is a positive valued threshold), then 4-pel MV is tested; otherwise, 4-pel MV is skipped. Basically, motion information and RD cost etc. are already known for ¼ pel MV when checking integer or 4-pel MV, which can be reused to speed up the encoding process of integer or 4-pel MV.

2.2.3 Examples of Affine Motion Compensation Prediction

(88) 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. In VVC, a simplified affine transform motion compensation prediction is applied with 4-parameter affine model and 6-parameter affine model. As shown in FIGS. **13**A and **13**B, the affine motion field of the block is described by two (in the 4-parameter affine model that uses the variables a, b, e and f) or three (in the 6-parameter affine model that uses the variables a, b, c, d, e and f) control point motion vectors, respectively.

(89) The motion vector field (MVF) of a block is described by the following equation with the 4-parameter affine model and 6-parameter affine model respectively:

$$
(90) \quad \left\{ \begin{array}{l} mv^h(x, y) = ax - by + e = \frac{(mv_1^h - mv_0^h)}{w}x - \frac{(mv_1^v - mv_0^v)}{w}y + mv_0^h \\ mv^v(x, y) = bx + ay + f = \frac{(mv_1^v - mv_0^v)}{w}x + \frac{(mv_1^h - mv_0^h)}{w}y + mv_0^v \end{array} \right. \quad \text{Eq.}(1)
$$

$$
\left\{ \begin{array}{l} mv^h(x, y) = ax + cy + e = \frac{(mv_1^h - mv_0^h)}{w}x + \frac{(mv_2^h - mv_0^h)}{h}y + mv_0^h \\ mv^v(x, y) = bx + dy + f = \frac{(mv_1^v - mv_0^v)}{w}x + \frac{(mv_2^v - mv_0^v)}{h}y + mv_0^v \end{array} \right. \quad \text{Eq.}(2)
$$

(91) Herein, (mv.sup.h.sub.0, mv.sup.h.sub.0) is motion vector of the top-left corner control point (CP), and (mv.sup.h.sub.1, mv.sup.h.sub.1) is motion vector of the top-right corner control point and (mv.sup.h.sub.2, mv.sup.h.sub.2) is motion vector of the bottom-left corner control point, (x, y) represents the coordinate of a representative point relative to the top-left sample within current block. The CP motion vectors may be signaled (like in the affine AMVP mode) or derived on-the-fly (like in the affine merge mode). w and h are the width and height of the current block. In practice, the division is implemented by right-shift with a rounding operation. In an example, the representative point is defined to be the center position of a sub-block, e.g., when the coordinate of the left-top corner of a sub-block relative to the top-left sample within current block is (xs, ys), the coordinate of the representative point is defined to be (xs+2, ys+2). For each sub-block (e.g., 4×4), the representative point is utilized to derive the motion vector for the whole sub-block.

(92) FIG. **14** shows an example of affine MVF per sub-block for a block **1400**, wherein in order to further simplify the motion compensation prediction, sub-block based affine transform prediction is applied. To derive motion vector of each M×N sub-block, the motion vector of the center sample of each sub-block can be calculated according to Eqs. (1) and (2), 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. The interpolation filters for 1/16-pel are introduced by the affine mode. 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.

2.2.4 Examples for Signaling of Affine Prediction

(93) Similar to the translational motion model, there are also two modes for signaling the side information due affine prediction. They are AFFINE_INTER and AFFINE_MERGE modes.

2.2.5 Examples of AF_INTER Mode

(94) 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.

(95) In this mode, for each reference picture list (List 0 or List 1), an affine AMVP candidate list is constructed with three types of affine motion predictors in the following order, wherein each candidate includes the estimated CPMVs of the current block. The differences of the best CPMVs found at the encoder side (such as mv.sub.0 mv.sub.1 , mv.sub.2 in FIG. **17**) and the estimated CPMVs are signalled. In addition, the index of affine AMVP candidate from which the estimated CPMVs are derived is further signalled.

(96) 1) Inherited Affine Motion Predictors

(97) The checking order is similar to that of spatial MVPs in HEVC AMVP list construction. First, a left inherited affine motion predictor is derived from the first block in {A1, A0} that is affine coded and has the same reference picture as in current block. Second, an above inherited affine motion predictor is derived from the first block in {B1, B0, B2} that is affine coded and has the same reference picture as in current block. The five blocks A1, A0, B1, B0, B2 are depicted in FIG. **16**.

(98) Once a neighboring block is found to be coded with affine mode, the CPMVs of the coding unit covering the neighboring block are used to derive predictors of CPMVs of current block. For example, if A1 is coded with non-affine mode and A0 is coded with 4-parameter affine mode, the left inherited affine MV predictor will be derived from A0. In this case, the CPMVs of a CU covering A**0**, as denoted by MV.sub.0.sup.N for the top-left CPMV and MV.sub.1.sup.N for the top-right CPMV in FIG. **18**B are utilized to derive the estimated CPMVs of current block, denoted by MV.sub.0.sup.C, MV.sub.1.sup.C, MV.sub.2.sup.C for the top-left (with coordinate (x0, y0)), top-right (with coordinate (x1, y1)) and bottom-right positions (with coordinate (x2, y2)) of current block.

(99) 2) Constructed Affine Motion Predictors

(100) A constructed affine motion predictor consists of control-point motion vectors (CPMVs) that are derived from neighboring inter coded blocks, as shown in FIG. **17**, that have the same reference picture. If the current affine motion model is 4-paramter affine, the number of CPMVs is 2, otherwise if the current affine motion model is 6-parameter affine, the number of CPMVs is 3. The top-left CPMV mv.sub.0 is derived by the MV at the first block in the group {A, B, C} that is inter coded and has the same reference picture as in current block. The top-right CPMV mv.sub.1 is derived by the MV at the first block in the group {D, E} that is inter coded and has the same reference picture as in current block. The bottom-left CPMV mv.sub.2 is derived by the MV at the first block in the group {F, G} that is inter coded and has the same reference picture as in current block. If the current affine motion model is 4-parameter affine, then a constructed affine motion predictor is inserted into the candidate list only if both mv.sub.0 and mv.sub.1 are founded, that is, mv.sub.0 and mv.sub.1 are used as the estimated CPMVs for top-left (with coordinate (x0, y0)), top-right (with coordinate (x1, y1)) positions of current block. If the current affine motion model is 6-parameter affine, then a constructed affine motion predictor is inserted into the candidate list only if mv.sub.0, mv.sub.1 and mv.sub.2 are all founded, that is, mv.sub.0, mv.sub.1 and mv.sub.2 are used as the estimated CPMVs for top-left (with coordinate (x0, y0)), top-right (with coordinate (x1 , y1)) and bottom-right (with coordinate (x2, y2)) positions of current block.

No pruning process is applied when inserting a constructed affine motion predictor into the candidate list.

3) Normal AMVP Motion Predictors

The following applies until the number of affine motion predictors reaches the maximum. 1) Derive an affine motion predictor by setting all CPMVs equal to mv.sub.2 if available. 2) Derive an affine motion predictor by setting all CPMVs equal to mv.sub.1if available. 3) Derive an affine motion predictor by setting all CPMVs equal to mv.sub.0if available. 4) Derive an affine motion predictor by setting all CPMVs equal to HEVC TMVP if available. 5) Derive an affine motion

predictor by setting all CPMVs to zero MV.

Note that mv.sub.i is already derived in constructed affine motion predictor.

(101) In AF_INTER mode, when 4/6 parameter affine mode is used, ⅔ control points are required, and therefore ⅔ MVD needs to be coded for these control points, as shown in FIGS. **15**A and **15**B. In an existing implementation, the MV may be derived as follows, e.g., it predicts mvd.sub.1 and mvd.sub.2 from mvd.sub.0.

mv.sub.0=mv.sub.0+mvd.sub.0

mv.sub.1=mv.sub.1+mvd.sub.1+mvd.sub.0

mv.sub.2=mv.sub.2+mvd.sub.2+mvd.sub.0

(102) Herein, mv.sub.i, mvd.sub.i, and mv.sub.1 are the predicted motion vector, motion vector difference and motion vector of the top-left pixel (i=0), top-right pixel (i=1) or left-bottom pixel (i=2) respectively, as shown in FIG. **15**B. In some embodiments, the addition of two motion vectors (e.g., mvA(xA, yA) and mvB(xB, yB)) is equal to summation of two components separately. For example, newMV=mvA+mvB implies that the two components of newMV are set to (xA+xB) and (yA+yB), respectively.

2.2.6 Examples of AF_MERGE Mode

(103) 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. And the selection order for the candidate block is from left, above, above right, left bottom to above left as shown in FIG. **18**A (denoted by A, B, C, D, E in order). For example, if the neighbour left bottom block is coded in affine mode as denoted by A0 in FIG. **18**B, the Control Point (CP) motion vectors mv.sub.0.sup.N, mv.sub.1.sup.N and mv.sub.2.sup.N of the top left corner, above right corner and left bottom corner of the neighbouring CU/PU which contains the block A are fetched. And the motion vector mv.sub.0.sup.C, mv.sub.1.sup.C and mv.sub.2.sup.C (which is only used for the 6-parameter affine model) of the top left corner/top right/bottom left on the current CU/PU is calculated based on mv.sub.0.sup.N, mv.sub.1.sup.N and mv.sub.2.sup.N. It should be noted that sub-block (e.g., 4×4 block) located at the top-left corner stores mv0, the sub-block located at the top-right corner stores mvl if the current block is affine coded. If the current block is coded with the 6-parameter affine model, the sub-block located at the bottom-left corner stores mv2; otherwise (with the 4-parameter affine model), LB stores mv2′. Other sub-blocks stores the MVs used for MC.

(104) After the CPMV of the current CU v0 and v1 are computed according to the affine motion model in Eqs. (1) and (2), 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.

(105) In some embodiments, an affine merge candidate list may be constructed with following steps: 1) Insert inherited affine candidates

(106) Inherited affine candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. In the common base, as shown in FIG. **19**, the scan order for the candidate positions is: A1, B1, B0, A0 and B2.

(107) After a candidate is derived, full pruning process is performed to check whether same candidate has been inserted into the list. If a same candidate exists, the derived candidate is discarded. 2) Insert constructed affine candidates

(108) If the number of candidates in affine merge candidate list is less than MaxNumAffineCand (set to 5 in this contribution), constructed affine candidates are inserted into the candidate list. Constructed affine candidate means the candidate is constructed by combining the neighbor motion information of each control point.

(109) The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in FIG. **19**. CPk (k=1, 2, 3, 4) represents the k-th control point. A0, A1, A2, B0, B1, B2 and B3 are spatial positions for predicting CPk (k=1, 2, 3); T is temporal position for predicting CP4.

(110) The coordinates of CP1, CP2, CP3 and CP4 is (0, 0), (W, 0), (H, 0) and (W, H), respectively, where W and H are the width and height of current block.

(111) The motion information of each control point is obtained according to the following priority order:

(112) For CP1, the checking priority is B2.fwdarw.B3.fwdarw.A2. B2 is used if it is available. Otherwise, if B2 is available, B3 is used. If both B2 and B3 are unavailable, A2 is used. If all the three candidates are unavailable, the motion information of CP1 cannot be obtained.

(113) For CP2, the checking priority is B1.fwdarw.B0.

(114) For CP3, the checking priority is A1.fwdarw.A0.

(115) For CP4, T is used.

(116) Secondly, the combinations of controls points are used to construct an affine merge candidate.

(117) Motion information of three control points are needed to construct a 6-parameter affine candidate. The three control points can be selected from one of the following four combinations ({CP1, CP2, CP4}, {CP1, CP2, CP3 }, {CP2, CP3, CP4}, {CP1, CP3, CP4}). Combinations {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4} will be converted to a 6-parameter motion model represented by top-left, top-right and bottom-left control points.

(118) Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the following six combinations ({CP1, CP4}, {CP2, CP3 }, {CP1, CP2}, {CP2, CP4}, {CP1, CP3 }, {CP3, CP4}). Combinations {CP1, CP4}, {CP2, CP3 }, {CP2, CP4}, {CP1, CP3 }, {CP3, CP4} will be converted to a 4-parameter motion model represented by top-left and top-right control points.

(119) The combinations of constructed affine candidates are inserted into to candidate list as following order:

(120) {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}, {CP2, CP3}, {CP1, CP4}, {CP2, CP4}, {CP3, CP4}

(121) For reference list X (X being 0 or 1) of a combination, the reference index with highest usage ratio in the control points is selected as the reference index of list X, and motion vectors point to difference reference picture will be scaled.

(122) After a candidate is derived, full pruning process is performed to check whether same candidate has been inserted into the list. If a same candidate exists, the derived candidate is discarded. 3) Padding with zero motion vectors

(123) If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are insert into the candidate list, until the list is full.

(124) In some embodiments, for the sub-block merge candidate list, a 4-parameter merge candidate with MVs set to (0, 0) and prediction direction set to uni-prediction from list 0 (for P slice) and bi-prediction (for B slice).

2.3 Examples of Intra Block Copy (IBC)

(125) Intra block copy (IBC, or intra picture block compensation), also named current picture referencing (CPR) was adopted in HEVC screen content coding extensions (SCC). This tool is very efficient for coding of screen content video in that repeated patterns in text and graphics rich content occur frequently within the same picture. Having a previously reconstructed block with equal or similar pattern as a predictor can effectively reduce the prediction error and therefore improve coding efficiency. An example of the intra block compensation is illustrated in FIG. **20**.

(126) Similar to the design of CRP in HEVC SCC, In VVC, The use of the IBC mode is signaled at both sequence and picture level. When the IBC mode is enabled at sequence parameter set (SPS), it can be enabled at picture level. When the IBC mode is enabled at picture level, the current reconstructed picture is treated as a reference picture. Therefore, no syntax change on block level is needed on top of the existing VVC inter mode to signal the use of the IBC mode.

(127) Some of the features of CPR include: It is treated as a normal inter mode. Therefore, merge

and skip modes are also available for the IBC mode. The merge candidate list construction is unified, containing merge candidates from the neighboring positions that are either coded in the IBC mode or the HEVC inter mode. Depending on the selected merge index, the current block under merge or skip mode can merge into either an IBC mode coded neighbor or otherwise a normal inter mode coded one with different pictures as reference pictures. Block vector prediction and coding schemes for the IBC mode reuse the schemes used for motion vector prediction and coding in the HEVC inter mode (AMVP and MVD coding). The motion vector for the IBC mode, also referred as block vector, is coded with integer-pel precision, but stored in memory in 1/16-pel precision after decoding as quarter-pel precision is required in interpolation and deblocking stages. When used in motion vector prediction for the IBC mode, the stored vector predictor will be right shifted by **4**. Search range: it is restricted to be within the current CTU. CPR is disallowed when affine mode/triangular mode/GBi/weighted prediction is enabled.

2.3.1 Harmonization with CPR

(128) In PCT/CN2018/089920, several methods are proposed to harmonize CPR (a.k.a., IBC) and some new coding tools.

2.4 Examples of Merge List Design in VVC

(129) There are three different merge list construction processes supported in VVC: (1) Sub-block merge candidate list: it includes ATMVP and affine merge candidates. One merge list construction process is shared for both affine modes and ATMVP mode. Here, the ATMVP and affine merge candidates may be added in order. Sub-block merge list size is signaled in slice header, and maximum value is 5. (2) Uni-Prediction TPM merge list: For triangular prediction mode, one merge list construction process for the two partitions is shared even two partitions could select their own merge candidate index. When constructing this merge list, the spatial neighbouring blocks and two temporal blocks of the block are checked. The motion information derived from spatial neighbours and temporal blocks are called regular motion candidates in our IDF. These regular motion candidates are further utilized to derive multiple TPM candidates. Please note the transform is performed in the whole block level, even two partitions may use different motion vectors for generating their own prediction blocks. In some embodiments, the uni-prediction TPM merge list size is fixed to be 5. (3) Regular merge list: For remaining coding blocks, one merge list construction process is shared. Here, the spatial/temporal/HMVP, pairwise combined bi-prediction merge candidates and zero motion candidates may be inserted in order. Regular merge list size is signaled in slice header, and maximum value is 6.

Sub-Block Merge Candidate List

(130) It is suggested that all the sub-block related motion candidates are put in a separate merge list in addition to the regular merge list for non-sub block merge candidates.

(131) The sub-block related motion candidates are put in a separate merge list is named as 'sub-block merge candidate list'.

(132) In one example, the sub-block merge candidate list includes affine merge candidates, and ATMVP candidate, and/or sub-block based STMVP candidate.

(133) In another example, the ATMVP merge candidate in the normal merge list is moved to the first position of the affine merge list. Such that all the merge candidates in the new list (i.e., sub-block based merge candidate list) are based on sub-block coding tools.

(134) In still another example, a special merge candidate list, known as sub-block merge candidate list (a.k.a., affine merge candidate list) is added besides the regular merge candidate list. The sub-block merge candidate list is filled with candidates in the following order: (a) ATMVP candidate (maybe available or unavailable); (b) Inherited Affine candidates; (c) Constructed Affine candidates; (d) Padding as zero MV 4-parameter affine model

(135) The maximum number of candidates (denoted as ML) in the sub-block merge candidate list derived as below: (1) If the ATMVP usage flag (e.g., the flag may be named as "sps_sbtmvp_enabled_flag") is on (equal to 1), but the affine usage flag (e.g., the flag may be

named as "sps_affine_enabled_flag") is off (equal to 0), then ML is set equal to 1. (2) If the ATMVP usage flag is off (equal to 0), and the affine usage flag is off (equal to 0), then ML is set equal to 0. In this case, the sub-block merge candidate list is not used. (3) Otherwise (the affine usage flag is on (equal to 1), the ATMVP usage flag is on or off), ML is signaled from the encoder to the decoder. Valid ML is $0 \leq ML \leq 5$.

(136) When constructing the sub-block merge candidate list, ATMVP candidate is checked first. If any one of the following conditions is true, ATMVP candidate is skipped and not put into the sub-block merge candidate list. (1) The ATMVP usage flag is off; (2) Any TMVP usage flag (e.g., the flag may be named as "slice_temporal_mvp_enabled_flag" when signaled at slice level) is off; (3) The reference picture with reference index 0 in reference list 0 is identical to the current picture (It is a CPR)

(137) ATMVP in the above described example is much simpler than in JEM. When an ATMVP merge candidate is generated, the following process is applied: (a) Check neighbouring blocks A1, B1, B0, A0 as shown in FIG. **2** in order, to find the first inter-coded, but not CPR-coded block, denoted as block X; (b) Initialize TMV=(0,0). If there is a MV (denoted as MVn) of block X, referring to the collocated reference picture (as signaled in the slice header), TMV is set equal to MVn. (c) Suppose the center point of the current block is $(x0, y0)$, then locate a corresponding position of $(x0,y0)$ as $M=(x0+MV*x, yO+MV*y)$ in the collocated picture. Find the block Z covering M. (i) If Z is intra-coded, then ATMVP is unavailable; (ii) If Z is inter-coded, MVZ_0 and MVZ_1 for the two lists of block Z are scaled to (Reflist 0 index 0) and (Reflist 1 index 1) as MVdefault0, MVdefault1, and stored. (d) For each 8×8 sub-block, suppose its center point is (xOS, yOS), then locate a corresponding position of $(x0S, y0S)$ as $MS=(x0S+MV*x, y0S+MV*y)$ in the collocated picture. Find the block ZS covering MS. (i) If ZS is intra-coded, MVdefault0, MVdefault1 are assigned to the sub-block; (ii) If ZS is inter-coded, MVZS_0 and MVZS_1 for the two lists of block ZS are scaled to (Reflist 0 index 0) and (Reflist 1 index 0) and are assigned to the sub-block.

(138) MV clipping and masking in ATMVP: When locating a corresponding position such as M or MS in the collocated picture, it is clipped to be inside a predefined region. The CTU size is S×S, S=128 in the present example. Suppose the top-left position of the collocated CTU is (xCTU, yCTU), then the corresponding position M or MS at (xN, yN) will be clipped into the valid region $xCTU \leq xN < xCTU+S+4$; $yCTU \leq yN < yCTU+S$. FIG. **21** shows an example of the valid region.

(139) Besides clipping, (xN, yN) is also masked as $xN=xN\&MASK$, $yN=yN\&MASK$, where MASK is an integer equal to $\sim(2.\text{sup}.N-1)$, and N=3, to set the lowest 3 bits to be 0. So xN and yN must be numbers which are times of 8. ("~" represents the bitwise complement operator).

(140) Regular Merge List

(141) Different from the merge list design, in VVC, the history-based motion vector prediction (HMVP) method is employed. In some embodiments, combined bi-predictive merge candidates have been replaced by pairwise bi-predictive merge candidates.

(142) HMVP

(143) In HMVP, the previously coded motion information is stored. The motion information of a previously coded block is defined as an HMVP candidate. Multiple HMVP candidates are stored in a table, named as the HMVP table, and this table is maintained during the encoding/decoding process on-the-fly. The HMVP table is emptied when starting coding/decoding a new slice. Whenever there is an inter-coded block, the associated motion information is added to the last entry of the table as a new HMVP candidate. The overall coding flow is depicted in FIG. **22**.

(144) HMVP candidates could be used in both AMVP and merge candidate list construction processes. FIG. **23** depicts the modified merge candidate list construction process (highlighted in grey). When the merge candidate list is not full after the TMVP candidate insertion, HMVP candidates stored in the HMVP table could be utilized to fill in the merge candidate list. Considering that one block usually has a higher correlation with the nearest neighbouring block in

terms of motion information, the HMVP candidates in the table are inserted in a descending order of indices. The last entry in the table is firstly added to the list, while the first entry is added in the end. Similarly, redundancy removal is applied on the HMVP candidates. Once the total number of available merge candidates reaches the maximal number of merge candidates allowed to be signaled, the merge candidate list construction process is terminated.

2.5 Examples of Generalized Bi-Prediction (GBi) Improvement

(145) Generalized Bi-prediction improvement (GBi) proposed in one example is provided.

(146) GBi was proposed in one example. An improved embodiment that improved the gain-complexity trade-off for GBi is provide which applies unequal weights to predictors from L0 and L1 in bi-prediction mode. In inter prediction mode, multiple weight pairs including the equal weight pair (½, ½) are evaluated based on rate-distortion optimization (RDO), and the GBi index of the selected weight pair is signaled to the decoder. In merge mode, the GBi index is inherited from a neighboring CU. In the embodiment, GBi, the predictor generation in bi-prediction mode is shown in the equation below.

$$P_{GBi}=(w0*PL0+w1*PL1+RoundingOffsetGBi)>>shiftNumGBi,$$

(147) Herein, PGBi is the final predictor of GBi. w0 and w1 are the selected GBi weight pair and applied to the predictors of list 0 (L0) and list 1 (L1), respectively. RoundingOffsetGBi and shiftNumGBi are used to normalize the final predictor in GBi. The supported w1 weight set is {−¼, ⅜, ½, ⅝, 5/4}, in which the five weights correspond to one equal weight pair and four unequal weight pairs. The blending gain, i.e., sum of w1 and w0, is fixed to 1.0. Therefore, the corresponding w0 weight set is { 5/4, ⅝, ½, ⅜, −¼}. The weight pair selection is at CU-level.

(148) For non-low delay pictures, the weight set size is reduced from five to three, where the w1 weight set is {⅜, ½, ⅝} and the w0 weight set is {⅝, ½, ⅜}. The weight set size reduction for non-low delay pictures is applied to the GBi and all the GBi tests in this contribution.

(149) In an embodiment, a combined solution is proposed to further improve the GBi performance. Specifically, the following modifications are applied on top of the existing GBi design in the embodiment.

(150) GBi Encoder Bug Fix

(151) To reduce the GBi encoding time, in current encoder design, the encoder will store uni-prediction motion vectors estimated from GBi weight equal to 4/8, and reuse them for uni-prediction search of other GBi weights. This fast encoding method is applied to both translation motion model and affine motion model. In one example, 6-parameter affine model was adopted together with 4-parameter affine model. The encoder does not differentiate 4-parameter affine model and 6-parameter affine model when it stores the uni-prediction affine MVs when GBi weight is equal to 4/8. Consequently, 4-parameter affine MVs may be overwritten by 6-parameter affine MVs after the encoding with GBi weight 4/8. The stored 6-parmater affine MVs may be used for 4-parameter affine ME for other GBi weights, or the stored 4-parameter affine MVs may be used for 6-parameter affine ME. The proposed GBi encoder bug fix is to separate the 4-paramerter and 6-parameter affine MVs storage. The encoder stores those affine MVs based on affine model type when GBi weight is equal to 4/8, and reuse the corresponding affine MVs based on the affine model type for other GBi weights.

(152) CU Size Constraint for GBi

(153) In this method, GBi is disabled for small CUs. In inter prediction mode, if bi-prediction is used and the CU area is smaller than 128 luma samples, GBi is disabled without any signaling.

(154) Merge Mode with GBi

(155) With Merge mode, GBi index is not signaled. Instead it is inherited from the neighbouring block it is merged to. When TMVP candidate is selected, GBi is turned off in this block.

(156) Affine Prediction with GBi

(157) When the current block is coded with affine prediction, GBi can be used. For affine inter mode, GBi index is signaled. For Affine merge mode, GBi index is inherited from the neighbouring

block it is merged to. If a constructed affine model is selected, GBi is turned off in this block.

2.6 Examples of Inter-Intra Prediction (IIP) Mode

(158) With inter-intra prediction mode, multi-hypothesis prediction combines one intra prediction and one merge indexed prediction. Such a block is treated as a special inter-coded block. In a merge CU, one flag is signaled for merge mode to select an intra mode from an intra candidate list when the flag is true. For luma component, the intra candidate list is derived from 4 intra prediction modes including direct current (DC), planar, horizontal, and vertical modes, and the size of the intra candidate list can be 3 or 4 depending on the block shape. When the CU width is larger than the double of CU height, horizontal mode is exclusive of the intra mode list and when the CU height is larger than the double of CU width, vertical mode is removed from the intra mode list. One intra prediction mode selected by the intra mode index and one merge indexed prediction selected by the merge index are combined using weighted average. For chroma component, direct mode (DM) is always applied without extra signaling.

(159) The weights for combining predictions are described as follow. When DC or planar mode is selected or the CB width or height is smaller than 4, equal weights are applied. For those CBs with CB width and height larger than or equal to 4, when horizontal/vertical mode is selected, one CB is first vertically/horizontally split into four equal-area regions. Each weight set, denoted as (w_intra_i, w_inter_i), where i is from 1 to 4 and $(w\_intra1, w\_inter1)=(6, 2)$, $(w\_intra2, w\_inter2)=(5, 3)$, $(w\_intra3, w\_inter3)=(3, 5)$, and $(w\_intra4, w\_inter4)=(2, 6)$, will be applied to a corresponding region. (w_intra1, w_inter1) is for the region closest to the reference samples and (w_intra4, w_inter4) is for the region farthest away from the reference samples. Then, the combined prediction can be calculated by summing up the two weighted predictions and right-shifting 3 bits. Moreover, the intra prediction mode for the intra hypothesis of predictors can be saved for reference of the following neighboring CUs.

(160) Detailed design in an example

(161) 1. inter_intra_flag (one new context) is signaled if all the conditions are satisfied a. Merge but not skip b. Not affine c. Not MMVD d. W*H<64 and W!=128 and H!=128 (That means we may have 2*8 chroma intra-prediction here) 2. Inter-intra prediction can be applied on triangular prediction; 3. Inter-intra prediction is applied to Y, Cb and Cr. 4. If inter_intra_flag is on, the intra-prediction mode (IPM) is signaled. IPM must be one of {DC, Planar, Ver, Hor} a. 3 MPMs are involved. b. If W>2*H or H>2*W, IPM must be one of the 3 MPMs, and MPM-flag is not signaled; Otherwise, MPM-flag is signaled (one new context, why not share the same context with intra?) i. If MPM_flag is on, signal the selected MPM. 0, 10, 11 represents the three MPMs (bypass) ii. Otherwise, find the one in {DC, Planar, Ver, Hor} which is not in MPM list, as the IPM. c. MPM list construction (different to Intra MPM list) i. Set LeftIPM=AboveIPM=DC; ii. If the left neighbouring block (A2) is intra-coded or inter-intra-coded, set leftIPM to the IPM of the left neighbouring block; iii. If the above neighbouring block (B3) is intra-coded or inter-intra-coded, set aboveIPM to the IPM of the above neighbouring block; iv. leftIPM =(leftIPM>DC_IDX)? ((leftIPM<DIA_IDX)? HOR_IDX: VER_IDX): leftIPM; v. aboveIPM=(aboveIPM>DC_IDX)? ((aboveIPM<DIA_IDX)? HOR_IDX: VER_IDX): aboveIPM; (What about the wide-angle modes?) vi. If leftIPM ==aboveIPM 1. If leftIPM is not DC or planar   a. MPM[0]=leftIPM, MPM[1]=Planar, MPM[2]=DC 2. Otherwise, a. MPM[0]=DC, MPM[1]=Planar, MPM[2]=Ver vii. Otherwise 1. MPM[0]=leftIPM, MPM[1]=aboveIPM 2. If non of leftIPM and aboveIPM is planar, MPM[2]=Planar 3. Otherwise: MPM[2]=(leftIPM+aboveIPM)<2? Ver: DC viii. If W>2*H 1. If MPM[x]==Hor, Find ModeX in {DC, Planar, Ver, Hor} but not in the MPM list, set MPM[x]=ModeX; ix. If H>2*W 1. If MPM[x]==Ver, Find ModeX in {DC, Planar, Ver, Hor} but not in the MPM list, set MPM[x]=ModeX; 5. It should be noted that the IPM used by inter-intra-prediction cannot be used to predict IPM of an intra-coded block. a. An intra-coded block treated an inter-intra coded block as inter-coded 6. Neighbouring sample filtering: As intra-prediction: a. No filtering for DC/VER/HOR b. Filtering for Planar when W*H>32; c. PDPC is kept 7. Weighting

sum method a. P(x,y)=Clip((W0(x,y)*P_Intra(x,y)+W1(x,y)*P_Inter(x,y))>>3) 8. Weighting values (may be improved or simplified) a. If mode==DC/Planar‖W<4‖H<4: W0=W1=4; b. If mode==HOR: i. W0=6, W1=2 if x<W/4; ii. W0=5, W1=3 if W/4≤x<W/2; iii. W0=3, W1=5 if W/2≤x<3*W/4; iv. W0=2, W1=6 if x≥3*W/4; c. If mode==VER: i. W0=6, W1=2 if y<H/4; ii. W0=5, W1=3 if H/4≤y<H/2; iii. W0=3, W1=5 if H/2≤y<3*H/4; iv. W0=2, W1=6 if y≥3*H/4.

2.7 Examples of Ultimate Motion Vector Expression (UMVE)

(162) In some embodiments, ultimate motion vector expression (UMVE) is presented. UMVE is used for either skip or merge modes with a proposed motion vector expression method.

(163) UMVE re-uses merge candidate as same as those included in the regular merge candidate list in VVC. Among the merge candidates, a base candidate can be selected, and is further expanded by the proposed motion vector expression method.

(164) UMVE provides a new motion vector difference (MVD) representation method, in which a starting point, a motion magnitude and a motion direction are used to represent a MVD.

(165) This proposed technique uses a merge candidate list as it is. But only candidates which are default merge type (MRG_TYPE_DEFAULT_N) are considered for UMVE's expansion.

(166) Base candidate index defines the starting point. Base candidate index indicates the best candidate among candidates in the list as follows.

(167) TABLE-US-00001 TABLE 1 Base candidate IDX Base candidate IDX 0 1 2 3 N.sup.th MVP 1.sup.st MVP 2.sup.nd MVP 3.sup.rd MVP 4.sup.th MVP

(168) If the number of base candidate is equal to 1, Base candidate IDX is not signaled.

(169) Distance index is motion magnitude information. Distance index indicates the pre-defined distance from the starting point information. Pre-defined distance is as follows:

(170) TABLE-US-00002 TABLE 2 Distance IDX Distance IDX 0 1 2 3 4 5 6 7 Pixel ¼- ½- 1- 2- 4- 8- 16- 32- distance pel pel pel pel pel pel pel pel

(171) Direction index represents the direction of the MVD relative to the starting point. The direction index can represent of the four directions as shown below.

(172) TABLE-US-00003 TABLE 3 Direction IDX Direction IDX 00 01 10 11 x-axis + − N/A N/A y-axis N/A N/A + −

(173) UMVE flag is signaled right after sending a skip flag or merge flag. If skip or merge flag is true, UMVE flag is parsed. If UMVE flag is equal to 1, UMVE syntaxes are parsed. But, if not 1, AFFINE flag is parsed. If AFFINE flag is equal to 1, that is AFFINE mode, but, if not 1, skip/merge index is parsed for skip/merge mode.

(174) Additional line buffer due to UMVE candidates is not needed. Because a skip/merge candidate of software is directly used as a base candidate. Using input UMVE index, the supplement of MV is decided right before motion compensation. There is no need to hold long line buffer for this.

(175) In current common test condition, either the first or the second merge candidate in the merge candidate list could be selected as the base candidate.

(176) UMVE is known as Merge with MVD (MMVD).

3 Drawbacks of Existing Implementations

(177) In existing implementations (e.g., current VVC design), CPR has these issues: (1) Interaction between CPR and affine is still unclear. (e.g., for CPR coded blocks, the merge candidates may be inserted to the sub-block merge list or regular merge list). (2) Interaction between CPR and general bi-prediction is still unclear. (e.g., for CPR coded blocks, always using fixed half-half weight, therefore, no need to signal weights index). (3) Interaction between CPR and MMVD (a.k.a., UMVE) is still unclear. (4) Interaction between CPR and inter-intra prediction is still unclear. (5) Interaction between ATMVP and CPR. (6) How to apply CPR with the dual (separate) coding tree is still unclear.

4 Example Methods for Motion Vector Derivation and Signaling for CPR

(178) Embodiments of the presently disclosed technology overcome the drawbacks of existing

implementations, thereby providing video coding with higher coding efficiencies. The use of motion vector derivation and signaling for current picture referencing in video and image coding, based on the disclosed technology, may enhance both existing and future video coding standards, is elucidated in the following examples described for various implementations. The examples of the disclosed technology provided below explain general concepts, and are not meant to be interpreted as limiting. In an example, unless explicitly indicated to the contrary, the various features described in these examples may be combined. It should be noted that "a block or a candidate uses CPR" may mean: (i) All reference pictures used by the block or candidate are CPR; or (ii) At least one reference picture used by the block or candidate is CPR.

(179) Example 1. CPR is not allowed to be used when affine motion compensation is used. (a) In one example, a conformance bit-stream cannot use CPR and affine motion compensation simultaneously in a single block. (b) In one example, in the parsing process, the block-level affine enabling/disabling flag may be firstly signaled. When one block is coded with the affine mode, there is no need to signal whether the current block's reference picture index is equal to the one that current picture is associated with. (c) Alternatively, reference picture indices may be signaled before the block-level affine enabling/disabling flag. When a block uses CPR, the signalling of affine flag is always skipped. (d) Alternatively, reference picture indices may be signaled before the block-level affine enabling/disabling flag. When a block is coded with bi-prediction and only one reference picture is the current reference picture, the affine flag may be still signalled. (i) In one example, furthermore, it is required that integer motion vectors are used for such a block.

(180) Example 2. A block cannot inherit affine model from a neighbouring block which uses CPR. (a) In one example, a neighbouring block using CPR cannot be used to derive an affine merge candidate in the sub-block merge candidate list. (i) In one example, a neighbouring block using CPR is treated as unavailable in the sub-block merge candidate list construction process. (b) In one example, a neighbouring block using CPR cannot be used to derive an affine AMVP candidate in the affine AMVP list. (i) In one example, a neighbouring block using CPR is treated as unavailable in the affine AMVP candidate list construction process.

(181) Example 3. Whether a block can inherit affine model from a neighbouring block which uses CPR or not, may depend on reference list. (a) In one example, suppose the neighbouring block applies bi-prediction, the reference picture in reference list X (X may be 0 or 1) is the current picture and the reference picture in reference list (1−X) is not the current picture, then (i) Motion information of the neighbouring block referring to reference list X cannot be used to derive an affine merge candidate in the sub-block merge candidate list. (1) In one example, motion information of the neighbouring block referring to reference list (1−X) may be used to derive an affine merge candidate in the sub-block merge candidate list. (ii) Motion information of the neighbouring block referring to reference list X cannot be used to derive an affine AMVP candidate in the affine AMVP list. (1) For example, motion information of the neighbouring block referring to reference list X cannot be used to derive an affine AMVP candidate in the affine AMVP list for reference list X. (iii) Motion information of the neighbouring block referring to reference list (1−X) may be used to derive an affine AMVP candidate in the affine AMVP list. (1) For example, Motion information of the neighbouring block referring to reference list (1−X) may be used to derive an affine AMVP candidate in the affine AMVP list for reference list (1−X).

(182) Example 4. CPR is not allowed to be used when GBi is applied (GBi weighting values are not equal on the two reference pictures) in a block. (a) In one example, a conformance bit-stream cannot use CPR and GBi simultaneously in a single block. (b) In one example, if CPR is used in a block, the GBi weighting index is not signaled and inferred to be 0 (weighting values are equal on the two reference pictures). (c) In one example, if un-equal weights in GBi is applied (GBi weighting values are not equal on the two reference pictures) in a block, the CPR indication is not signaled and inferred to be 0 (CPR is not used).

(183) Example 5. CPR is not allowed to be used when MMVD is applied in a block. (a) In one

example, a conformance bit-stream cannot use CPR and MMVD simultaneously in a single block. (b) In one example, if CPR is used in a block, MMVD flag is not signaled and inferred to be 0 (weighting values are equal on the two reference pictures). (c) In one example, if MMVD is applied in a block, the CPR indication is not signaled and inferred to be 0 (CPR is not used).

(184) Example 6. CPR is allowed to be used when MMVD is applied in a block. (a) In one example, if the base MV candidate has at least one reference pictures referring to the current picture: (i) In one example, MMVD is based on the MV referring to the current picture, omitting the MV not referring to the current picture (if existing); (ii) In one example, the distances signaled by MMVD can only be integers. (iii) In one example, MMVD is uni-predicted. (b) In one example, if the base MV candidate has at least one reference pictures not referring to the current picture: (i) In one example, MMVD is based on the MV referring not to the current picture, omitting the MV referring to the current picture (if existing);

(185) Example 7. CPR is not allowed to be used when inter-intra prediction is applied in a block. (a) In one example, a conformance bit-stream cannot use CPR and inter-intra prediction simultaneously in a single block. (b) In one example, if CPR is used in a block, the inter-intra prediction flag is not signaled and inferred to be 0. (c) In one example, if inter-intra prediction is applied in a block, the CPR indication is not signaled and inferred to be 0 (CPR is not used).

(186) Example 8. CPR is allowed to be used when inter-intra prediction is applied in a block. (a) In one example, if the merge candidate to do inter-intra prediction has at least one reference pictures referring to the current picture: (i) In one example, inter-intra prediction is based on the MV referring to the current picture, omitting the MV not referring to the current picture (if existing); (ii) In one example, the intra-prediction part of the inter-intra-prediction may be changed as: (1) Boundary filtering is not conducted; (2) Position Dependent Prediction Combination (PDPC) is not conducted; (3) Only these specific modes are allowed: MPMs DC Vertical Horizontal (iii) In one example, the inter-prediction part of the inter-intra-prediction is uni-predicted. (b) In one example, if the merge candidate to do inter-intra prediction has at least one reference pictures not referring to the current picture: (i) In one example, inter-intra prediction is based on the MV referring not to the current picture, omitting the MV referring to the current picture (if it exists).

(187) Example 9. In the process of ATMVP: (a) A neighbouring block is treated as unavailable if it uses CPR; (b) If a block in the collocated picture uses CPR (i) It may be treated as unavailable in the ATMVP process; (ii) It may be treated as available and a MV′ can be derived from that block for the current block or sub-block. (1) In one example, the MV′ refers to the current picture and is copied from the block in the collocated picture.

(188) Example 10. When CPR is coded as a new mode in addition to intra, inter modes (instead of being indicated as a specific reference picture in a reference picture list), whether to skip signalling of the CPR mode may depend on the decoded information of other coding methods. (a) In one example, the other coding methods may include, e.g., affine flag, inter-intra prediction flag, GBi index representing un-equal weights, MMVD, ATMVP. (b) When any one of the other coding method is applied to one block, the signalling of the CPR mode for this block is skipped. (c) Alternatively, whether to skip signalling of usage of other coding methods may depend on the decoded information of CPR mode.

(189) Example 11. When the luma component and chroma components are coded with separated coding structure trees: (a) In one example, when chroma components are coded, MV information (MV may refer to the current picture) of the corresponding block of the luma component can be used as a MV prediction or merge candidate. (i) The corresponding block of the luma component may be at any position (such as top-left, top-right, bottom-left, bottom-right or center) in the current chroma block. (1) If the colour format is not 4:4:4, the current chroma block should be scaled to find the corresponding luma block. (ii) In one example, there may be MV information from several corresponding blocks of the luma component severing as MV prediction or merge candidate for the current chroma block. Which one of them is selected may be signaled to the

decoder or derived at the decoder. (b) The width/height of a chroma block using CPR may have restrictions. (i) In one example, the width/height of a chroma block using CPR may be restricted differently from an intra-coded chroma block. (ii) In one example, width>2 && height>2 for intra coded block but width≥2 && height≥2 for CPR coded.

(190) The examples described above may be incorporated in the context of the method described below, e.g., methods **2400**, **2500**, **2600** and **2700**, which may be implemented at a video decoder or a video encoder.

(191) FIG. **24** shows a flowchart of an exemplary method for video processing. The method **2400** includes, at step **2410**, performing a conversion between a current video block and a bitstream representation of the current video block consistent with a first coding rule that specifies applicability of current picture referencing (CPR) during coding or decoding of the current video block, wherein the first coding rule specifies at least one of (a) refraining from simultaneously using CPR and affine motion compensation during the conversion, or (b) disabling, for the conversion, inheriting an affine model of a neighboring block that uses CPR for coding, or (c) inheriting, for the conversion, the affine model of the neighboring block only in case that a coding condition is met, or (d) refraining from simultaneously using CPR and generalized bi-prediction (GBi) for the conversion, wherein the GBi uses non-equally weighted reference pictures, or (e) refraining from simultaneously using CPR and a merge with motion vector differences (MMVD) mode, or (0 simultaneously using CPR and the MMVD mode, or (g) refraining from simultaneously using CPR and inter-intra prediction, or (h) simultaneously using CPR and inter-intra prediction, or (i) simultaneously using an alternative temporal motion vector predictor.

(192) In some embodiments, and in the context of Example 1, integer precision is used for coding rule (a).

(193) In some embodiments, and in the context of Example 6, the MMVD mode comprises a uni-prediction for coding rules (e) or (f).

(194) In some embodiments, and in the context of Example 8, the inter-intra prediction comprises an intra-prediction part and an inter-prediction part for coding rules (g) and (h). In an example, the intra-prediction part excludes boundary filtering. In another example, the intra-prediction part excludes position dependent prediction combination. In yet another example, the intra-prediction part comprises most probable mode selection, a DC prediction mode, a horizontal prediction mode or a vertical prediction mode. In yet another example, the inter-prediction part comprises uni-prediction.

(195) FIG. **25** shows a flowchart of another exemplary method for video processing. The method **2500** includes, at step **2510**, determining that a neighboring block of a current video block is coded using a current picture referencing mode.

(196) The method **2500** includes, at step **2520**, deriving, based on the determining, a merge candidate. In some embodiments, and in the context of Example 2, the merge candidate comprises a first affine model different from a second affine model of the neighboring block. In an example, the merge candidate comprises an affine merge candidate, and wherein the merge candidate list comprises a sub-block merge candidate list. In another example, the merge candidate comprises an affine AMVP candidate, and wherein the merge candidate list comprises an affine AMVP candidate list.

(197) In some embodiments, and in the context of Example 9, the merge candidate may include an alternative temporal motion vector prediction (ATMVP) candidate.

(198) The method **2500** includes, at step **2530**, inserting the merge candidate into a merge candidate list.

(199) The method **2500** includes, at step **2540**, reconstructing, based on the merge candidate list, the current video block.

(200) In some embodiments, and in the context of Example 3, the neighboring block applies bi-prediction, and the method **2500** further includes the step of determining, based on a first reference

picture list or a second reference picture list, an affine model of the merge candidate. In an example, the first reference picture list comprises a current picture of the current video block, and the affine model is determined based on at least one picture in the second reference picture list. In another example, the merge candidate comprises an AMVP candidate, and the merge candidate list comprises an affine AMVP candidate list.

(201) FIG. **26** shows a flowchart of yet another exemplary method for video processing. The method **2600** includes, at step **2610**, generating decoded information by decoding a first coding mode of a current video block that is coded with the first coding mode and a second coding mode that is different from the first coding mode.

(202) The method **2600** includes, at step **2620**, determining, based on the decoded information, whether the first coding mode or the second coding mode is signaled in a bitstream representation of the current video block.

(203) The method **2600** includes, at step **2630**, reconstructing, based on the determining, the current video block.

(204) In some embodiments, and in the context of Example 10, the first coding mode comprises an affine coding mode, an inter-intra prediction mode, a generalized bi-prediction prediction mode with unequal weights, a merge with motion vector differences mode or an alternative temporal motion vector prediction mode, the second coding mode comprises a CPR mode, and the bitstream representation excludes a flag for the CPR mode.

(205) In some embodiments, and in the context of Example 10, first coding mode comprises a current picture referencing (CPR) mode, the second coding mode comprises an affine coding mode, an inter-intra prediction mode, a generalized bi-prediction prediction mode with unequal weights, a merge with motion vector differences mode or an alternative temporal motion vector prediction mode, and the bitstream representation excludes a flag for the second coding mode.

(206) FIG. **27** shows a flowchart of yet another exemplary method for video processing. The method **2700** includes, at step **2710**, determining that a current video block comprises a luma component that is coded using a first coding structure tree and a chroma component that is coded using a second coding structure tree different from the first coding structure tree.

(207) The method **2700** includes, at step **2720**, reconstructing, based on the determining, the current video block. In some embodiments, and in the context of Example 11, the reconstructing comprises decoding the chroma component based on motion vector information that is derived from one or more corresponding blocks of the luma component. In an example, the one or more corresponding blocks of the luma component is in a top-left position, a top-right position, a bottom-left position, a bottom-right position or a center position of the current video block. In another example, the method **2700** further includes the step of scaling, upon determining that a color format of the chroma component is different from a 4:4:4 color format, the current video block to determine the one or more corresponding blocks of the luma component.

(208) In some embodiments, and in the context of Example 11, the current video block is coded using a current picture referencing (CPR) mode. In an example, a first set of restrictions on a height and a width of the current video block is different from a second set of restrictions on a height and a width of an intra-coded video block different from the current video block.

5 Example Implementations of the Disclosed Technology

(209) FIG. **28** is a block diagram of a video processing apparatus **2800**. The apparatus **2800** may be used to implement one or more of the methods described herein. The apparatus **2800** may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus **2800** may include one or more processors **2802**, one or more memories **2804** and video processing hardware **2806**. The processor(s) **2802** may be configured to implement one or more methods (including, but not limited to, methods **2400**, **2500**, **2600** and **2700**) described in the present disclosure. The memory (memories) **2804** may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware **2806**

may be used to implement, in hardware circuitry, some techniques described in the present disclosure.

(210) FIG. **29** shows a flowchart of yet another exemplary method for video processing. The method **2900** includes, at step **2910**, determining that an Intra-Block-Copy (IBC) mode is applied to a current video block of a video, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture where the current video block is located in; at step **2920**, making a decision regarding a disabling of a specific coding mode for the current block; at step **2930**, performing, based on the decision, a conversion between the current video block and the bitstream representation, wherein the specific coding mode uses a motion vector and a non-current picture to derive a prediction of a video block.

(211) FIG. **30** shows a flowchart of yet another exemplary method for video processing. The method **3000** includes, at step **3010**, determining that a current video block is coded using a specific coding mode; at step **3020**, making a decision regarding disabling of an Intra-Block-Copy (IBC) mode for the current block based on the determination, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture where the current block is located in; at step **3030**, performing, based on the decision, a conversion between the current block and the bitstream representation, wherein the specific coding mode uses a motion vector and a non-current picture to derive a prediction of the current block.

(212) FIG. **31** shows a flowchart of yet another exemplary method for video processing. The method **3100** includes, at step **3110**, determining an affine mode of a current video block, wherein whether the current video block inherits an affine mode from a neighbouring block depends on a reference list; at step **3120**, performing a conversion between a current video block of a video and a bitstream representation of the video consistent based on the determination.

(213) FIG. **32** shows a flowchart of yet another exemplary method for video processing. The method **3200** includes, at step **3210**, performing a conversion between a current video block of a video and a bitstream representation of the video consistent, wherein during the conversion an Intra-Block-Copy (IBC) mode and a merge with motion vector difference (MMVD) mode are used, and wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in, and in the MMVD mode, a motion vector of a video block is derived based on a merge motion candidate list and is further refined by at least one motion vector difference.

(214) FIG. **33** shows a flowchart of yet another exemplary method for video processing. The method **3300** includes, at step **3310**, performing a conversion between a current video block of a video and a bitstream representation of the video, wherein during the conversion an Intra-Block-Copy (IBC) mode and an inter-intra prediction mode are used, and wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in, and in the inter-intra prediction mode, prediction signal of the current video block is generated at least based on an intra prediction signal and an inter prediction signal.

(215) FIG. **34** shows a flowchart of yet another exemplary method for video processing. The method **3400** includes, at step **3410**, determining, during a conversion between a current video block and a bitstream representation of the current video block, decoded information of at least one coding method different from an IBC mode, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in; at step **3420**, determining, based on the decoded information, whether to skip signaling of an IBC flag for the current video block in a bitstream representation of the current video block, wherein the IBC flag relates with the IB mode; and at step **3430**, performing the conversion based on the determining.

(216) FIG. **35** shows a flowchart of yet another exemplary method for video processing. The method **3500** includes, at step **3510**, during a conversion between a current video block and a bitstream representation of the current video block, wherein the current video block comprises a

luma component that is coded using a first coding structure tree and a chroma component that is coded using a second coding structure tree different from the first coding structure tree, deriving first motion vector information for a chroma block from second motion vector information of one or more corresponding blocks of the luma component of the current video block; at step **3520**, performing the conversion based on the first motion vector information and the second vector information.

(217) FIG. **36** shows a flowchart of yet another exemplary method for video processing. The method **3600** includes, at step **3610**, determining whether an Intra Block Copy (IBC) mode, in which at least one reference picture used by the current video block is identical to a picture in which the current video block is located, is applied for the current video block; at step **3620**, signaling whether the IBC mode is applied for the current video block; at step **3630**, refraining from signaling flags for a group of prediction modes based on determining that the IBC mode is applied for the current video block; and at step **3640**, performing the conversion for the current video block based on the application of the IBC mode.

(218) FIG. **37** shows a flowchart of yet another exemplary method for video processing. The method **3700** includes, at step **3710**, parsing a bitstream to determining whether an Intra Block Copy (IBC) mode, in which at least one reference picture used by the current video block is identical to a picture in which the current video block is located, is applied for the current video block; at step **3720**, performing the conversion for the current video block based on the application of the IBC mode, wherein flags for a group of prediction modes are excluded from the bitstream based on determining that the CPR is applied for the current video block.

(219) FIG. **38** shows a flowchart of yet another exemplary method for video processing. The method **3800** includes, at step **3810**, in a process of acquiring motion information of a current video block, during a conversion between the current video block and a bitstream representation of the current video block, wherein the motion information of the current video block is based on at least one affine model of at least one neighboring block of the current video block; at step **3820**, determining whether a neighboring block of the current video block uses an Intra-Block Copy(IBC) mode in which at least one reference picture used by the neighboring video block is identical to a picture in which the neighboring block is located; at step **3830**, disabling, for the current video block, to derive an affine candidate from the neighboring block based on determining that the neighbouring block uses the IBC mode; and at step **3830**, performing the conversion based on the motion information of the current video block.

(220) FIG. **39** shows a flowchart of yet another exemplary method for video processing. The method **3900** includes, at step **3910**, splitting a current video block into sub-blocks, during a conversion between a current video block and a bitstream representation of the current video block based on subblock-based temporal motion vector prediction (SbTMVP) mode; at step **3920**, identifying, based on motion information of a spatial neighboring block of the current video block, a corresponding block, for a sub-block, in a motion source picture with a temporal vector; at step **3930**, generating, based on coded mode information of the corresponding block, a motion vector and a reference index of the sub-block from the corresponding block of the sub-block; at step **3940**, performing the conversion based on the generated motion vector and reference index.

(221) FIG. **40** shows a flowchart of yet another exemplary method for video processing. The method **4000** includes, at step **4010**, splitting a current video block into sub-blocks, during a conversion between a current video block and a bitstream representation of the current video block based on subblock-based temporal motion vector prediction (SbTMVP) mode; at step **4020**, identifying, based on coded mode information of a spatial neighboring block of the current video block, a corresponding block, for a sub-block, in a motion source picture with a temporal vector; at step **4030**, generating a motion vector and a reference index of the sub-block from the corresponding block of the sub-block; at step **4040**, performing the conversion based on the generated motion vector and reference index.

(222) In some embodiments, the video coding methods may be implemented using an apparatus that is implemented on a hardware platform as described with respect to FIG. **28**.

(223) Some embodiments may be described using the following examples. 1.1. A method for video processing, comprising: determining that an Intra-Block-Copy (IBC) mode is applied to a current video block of a video, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture where the current video block is located in; making a decision regarding a disabling of a specific coding mode for the current block; performing, based on the decision, a conversion between the current video block and the bitstream representation; wherein the specific coding mode uses a motion vector and a non-current picture to derive a prediction of a video block. 1.2. The method of example 1.1, wherein the conversion comprises encoding the current video block into the bitstream representation. 1.3. The method of example 1.1, wherein the conversion comprises decoding the bitstream representation of the current video block to generate pixel values of the current video block. 1.4. The method of any of examples 1-3, wherein the specific coding mode comprises an affine motion compensation mode, wherein the affine motion compensation mode uses at least one control point motion vector. 1.5. The method of example 1.4, wherein reference picture indices is signaled, and the signaling of an affine flag is skipped in response to the current video block using the IBC mode. 1.6. The method of example 1.4, the method further comprises: before the affine flag is signaled, signaling reference picture indices and determining whether bi-prediction mode is used for the current video block, wherein if bi-prediction mode is used for the current video block and only one reference picture is the current reference picture, then the affine flag is signaled; and if bi-prediction mode is used for the current video block and both reference pictures are the current reference picture, then the affine flag is not signaled. 1.7. The method of example 1.6, wherein integer precision is used for motion vectors of the current video block. 1.8. The method of any of examples 1.1-1.7, wherein the specific coding mode comprises a GBi mode, wherein in the GBi mode, different weighting values relates with different reference pictures. 1.9. The method of example 1.8, wherein a GBi weighting index is not signaled in response to the IBC mode being used in the current block. 1.10 The method of example 1.8, wherein the GBi weighting index is inferred to be 0. 1.11. The method of any of examples 1.1-1.10, wherein the specific coding mode comprises a merge with motion vector difference (MMVD) mode, wherein in the MMVD mode, a motion vector of a video block is derived based on a merge motion candidate list and is further refined by at least one motion vector difference. 1.12. The method of example 1.11, wherein an MMVD flag is not signaled in response to the IBC mode being used in the current video block. 1.13. The method of example 1.12, wherein the MMVD flag is inferred to be zero. 1.14. The method of any of examples 1.1-1.13, the specific coding mode comprises a combined inter-intra prediction (CIIP) mode, wherein in the CIIP mode, a prediction signal of the current video block is generated at least based on an intra prediction signal and an inter prediction signal. 1.15. The method of example 1.14, wherein a CIIP flag for the current video block is not signaled in response to the IBC mode being used in the current block. 1.16. The method of example 1.15, wherein the CIIP flag is inferred to be zero. 1.17. A method for video processing, comprising: determining that a current video block is coded using a specific coding mode; making a decision regarding disabling of an Intra-Block-Copy (IBC) mode for the current block based on the determination, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture where the current block is located in; performing, based on the decision, a conversion between the current block and the bitstream representation; wherein the specific coding mode uses a motion vector and a non-current picture to derive a prediction of the current block. 1.18. The method of example 1.17, wherein the conversion comprises encoding the current video block into the bitstream representation. 1.19. The method of example 1.17, wherein the conversion comprises decoding the bitstream representation of the current video block to generate pixel values of the current video block. 1.20. The method of any one of examples 1.1-1.3, wherein the specific coding mode comprises an affine motion compensation mode, wherein the

affine motion compensation uses at least one control point motion vector. 1.21. The method of example 1.20, wherein an affine flag in block level is signaled in the bitstream indicating whether the affine motion compensation mode is enabled or not for the current video block, and if the affine flag indicates that the affine motion compensation mode is enabled for the current video block, then it is not signaled whether the IBC mode is applied. 1.22. The method of any one of examples 1.17-1.19, wherein the specific coding mode comprises a GBi mode, wherein in the GBi mode, different weighting values relates with different reference pictures. 1.23. The method of example 1.22, wherein an IBC indication is not signaled in response to the GBi mode being applied. 1.24. The method of example 1.23, wherein the IBC indication is inferred to be 0. 1.25. The method of any one of examples 1.17-1.24, wherein the specific coding mode comprises a merge with motion vector difference (MMVD) mode, wherein in the MMVD mode, a motion vector of the current block is derived based on a merge motion candidate list and is further refined by at least one motion vector difference. 1.26. The method of example 1.25, wherein an IBC indication is not signaled in response to the MMVD mode being applied in the current video block. 1.27. The method of example 1.26, wherein the IBC indication is inferred to be zero. 1.28. The method of any of examples 1.17-1.27, the specific coding mode comprises a combined inter-intra prediction (CIIP) mode, wherein in the CIIP mode, a prediction signal of the current video block is generated at least based on an intra prediction signal and an inter prediction signal. 1.29. The method of example 1.28, wherein an IBC flag for the current video block is not signaled in response to the inter-intra prediction mode being applied. 1.30. The method of example 1.29, wherein the IBC flag is inferred to be zero. 1.31. A method for video processing, comprising: determining an affine mode of a current video block, wherein whether the current video block inherits an affine mode from a neighbouring block depends on a reference list; performing a conversion between a current video block of a video and a bitstream representation of the video consistent based on the determination. 1.32 The method of example 1.31, wherein the neighbouring block applies bi-prediction, a reference picture in a reference list X is the current picture and a reference picture in a reference list $(1-X)$ is not the current picture, wherein X is 0 or 1. 1.33. The method of example 1.32, wherein motion information of the neighbouring block referring to the reference list X is not used to derive an affine merge candidate in a sub-block merge candidate list. 1.34. The method of example 1.32 or 1.33, wherein motion information of the neighbouring block referring to the reference list $(1-X)$ is used to derive the affine merge candidate in the sub-block merge candidate list. 1.35. The method of example 1.32, wherein motion information of the neighbouring block referring to the reference list X is not used to derive an affine AMVP candidate in an affine AMVP list. 1.36. The method of example 1.35, wherein the motion information of the neighbouring block referring to the reference list X is not used to derive the affine AMVP candidate in the affine AMVP list for the reference list X. 1.37. The method of example 1.32, wherein motion information of the neighbouring block referring to the reference list $(1-X)$ is used to derive an affine AMVP candidate in an affine AMVP list. 1.38. The method of example 1.37, wherein the motion information of the neighbouring block referring to the reference list $(1-X)$ is used to derive the affine AMVP candidate in the affine AMVP list for the reference list $(1-X)$. 1.39. A method for video processing, comprising: performing a conversion between a current video block of a video and a bitstream representation of the video consistent; wherein during the conversion an Intra-Block-Copy (IBC) mode and a merge with motion vector difference (MMVD) mode are used, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in, and in the MMVD mode, a motion vector of a video block is derived based on a merge motion candidate list and is further refined by at least one motion vector difference. 1.40. The method of example 1.39, wherein, the MMVD mode uses a base MV candidate which has at least one reference picture referring to the current picture. 1.41. The method of example 1.40, wherein the MMVD is based on at least one MV referring to the current picture, and an MV not referring to the current picture is omitted if existing. 1.42. The method of example 1.41, wherein

distances signaled by the MMVD are integers. 1.43. The method of example 1.41, wherein the MMVD is uni-predicted. 1.44. The method of example 1.39, wherein if the MMVD mode uses a base MV candidate which has at least one reference pictures not referring to the current picture, the MMVD is based on at least one MV referring not to the current picture, and an MV referring to the current picture is omitted if existing. 1.45. A method for video processing, comprising: performing a conversion between a current video block of a video and a bitstream representation of the video; wherein during the conversion an Intra-Block-Copy (IBC) mode and an inter-intra prediction mode are used, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in, and in the inter-intra prediction mode, prediction signal of the current video block is generated at least based on an intra prediction signal and an inter prediction signal. 1.46. The method of example 1.45, wherein a merge candidate to be inter-intra predicted has at least one reference picture referring to the current picture. 1.47. The method of example 1.46, wherein the inter-intra prediction is based on at least one MV referring to the current picture, and an MV not referring to the current picture is omitted if existing. 1.48. The method of example 1.45, wherein an intra-prediction part of the inter-intra prediction mode excludes boundary filtering. 1.49. The method of example 1.45, wherein an intra-prediction part of the inter-intra prediction mode excludes position dependent prediction combination (PDPC). 1.50. The method of example 1.45, wherein an intra-prediction part of the inter-intra prediction mode comprises at least one of: most probable mode selection (MPMs), a DC prediction mode, a horizontal prediction mode or a vertical prediction mode. 1.51. The method of example 1.45, wherein an inter-prediction part of the inter-intra prediction mode is uni-predicted. 1.52. The method of any of examples 1.45-1.51, wherein if a merge candidate to be inter-intra predicted has at least one reference pictures not referring to the current picture, the inter-intra prediction based on at least one MV referring not to the current picture is performed and the MV referring to the current picture is omitted if existing. 1.53. A method for video processing, comprising: determining, during a conversion between a current video block and a bitstream representation of the current video block, decoded information of at least one coding method different from an IBC mode, wherein in the IBC mode, at least one reference picture used by the current video block is a current picture in which the current video block is located in; determining, based on the decoded information, whether to skip signaling of an IBC flag for the current video block in a bitstream representation of the current video block, wherein the IBC flag relates with the IB mode; and performing the conversion based on the determining. 1.54. The method of example 1.53, wherein the conversion comprises encoding the current video block into the bitstream representation. 1.55. The method of example 1.53, wherein the conversion comprises decoding the bitstream representation of the current video block current video block to generate pixel values of the current video block. 1.56. The method of any of examples 1.53-1.55, further comprising: if it is determined not to skip signaling of the IBC flag in the determining step, signaling the IBC flag for the current video block in the bitstream representation; otherwise, refraining from signaling the IBC flag for the current video block in the bitstream representation. 1.57. The method of any of examples 1.53-1.56, wherein it is determined to skip signaling of the IBC flag when the decoded information of coding method comprises at least one of an affine flag, an inter-intra prediction flag, a generalized bi-prediction prediction (GBi) index, a merge with motion vector differences (MMVD) information, an alternative temporal motion vector prediction (ATMVP) information. 1.58. A method for video processing, comprising: during a conversion between a current video block and a bitstream representation of the current video block, wherein the current video block comprises a luma component that is coded using a first coding structure tree and a chroma component that is coded using a second coding structure tree different from the first coding structure tree, deriving first motion vector information for a chroma block from second motion vector information of one or more corresponding blocks of the luma component of the current video block; performing the conversion based on the first motion vector information and the second vector information. 1.59.

The method of example 1.58, wherein the conversion comprises encoding the current video block into the bitstream representation. 1.60. The method of example 1.58, wherein the conversion comprises decoding the bitstream representation of the current video block current video block to generate pixel values of the current video block. 1.61. The method of example 1.58, wherein the first motion vector information is used as a MV predication or a merge candidate to derive the second motion vector information. 1.62. The method of and of examples 1.58-1.61, wherein the one or more corresponding blocks of the luma component are in a top-left position, a top-right position, a bottom-left position, a bottom-right position and/or a center position of the current video block. 1.63. The method of example 1.62, further comprising: determining whether a color format of the chroma component is different from a 4:4:4 color format; and scaling, upon determining that the color format of the chroma component is different from a 4:4:4 color format, the current video block to determine the one or more corresponding blocks of the luma component. 1.64. The method of any of examples 1.58-1.60, wherein the current video block is coded using an Intra-Block-Copy (IBC) mode. 1.65. The method of example 1.64, wherein a first set of restrictions on a height and a width of the current video block is different from a second set of restrictions on a height and a width of an intra-coded video block different from the current video block. 1.66. The method of example 1.65, wherein the width of the current video block is greater than 2 and the height is greater than 2 when the current video block is coded using an intra block copy mode, and the width of the current video block is greater or equal to 2 and the height is greater or equal to 2 when the current video block is coded using the IBC mode. 1.67. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of examples 1.1 to 1.66. 1.68. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples 1.1 to 1.66. 2.1. A method for video processing, comprising: determining whether an Intra Block Copy (IBC) mode, in which at least one reference picture used by the current video block is identical to a picture in which the current video block is located , is applied for the current video block; signaling whether the IBC mode is applied for the current video block; refraining from signaling flags for a group of prediction modes based on determining that the IBC mode is applied for the current video block; and performing the conversion for the current video block based on the application of the IBC mode. 2.2. The method of example 2.1, wherein the conversion comprises encoding the current video block into the bitstream representation. 2.3. The method of example 2.1 or 2.2, wherein the group of prediction modes include an affine mode, wherein the affine mode uses at least one control point motion vector. 2.4. The method of example 2.3, wherein the determining whether the IBC mode is applied for the current video block takes place before an affine flag is signaled; and the method further comprises: if it is determined that the IBC mode is applied for the current video block, then refraining from signaling the affine flag; and otherwise, signaling the affine flag. 2.5. The method of example 2.4, wherein when the affine flag is not signaled, the affine flag is inferred to be zero. 2.6. The method of any of examples 2.1-2.5, wherein, the group of prediction modes include a generalized bi-prediction (GBi) mode, wherein in the GBi mode, different weighting values relates with different reference pictures. 2.7. The method of example 2.6, wherein the determining whether the IBC mode is applied for the current video block takes place before a GBi index is signaled; and the method further comprises: if it is determined that the IBC mode is applied for the current video block, then refraining from signaling GBi weighting index, and otherwise, signaling the GBi weighting index. 2.8. The method of example 2.7, wherein when the GBi weighting index is not signaled, the weighting values of the reference pictures used by the GBi are inferred to be zero. 2.9. A method for video processing, comprising: parsing a bitstream to determining whether an Intra Block Copy (IBC) mode, in which at least one reference picture used by the current video block is identical to a picture in which the current video block is located, is applied for the current video block; performing the conversion for the current

video block based on the application of the IBC mode, wherein flags for a group of prediction modes are excluded from the bitstream based on determining that the CPR is applied for the current video block. 2.10. The method of example 2.9, wherein the conversion comprises decoding the bitstream representation of the current video block current video block to generate pixel values of the current video block. 2.11. The method of example 2.9 or 2.10, wherein the flag for the prediction mode is inferred to be zero. 2.12. The method of any of examples 2.9-2.11, wherein the group of prediction modes include at least one of: an affine mode in which at least one control point motion vector is used, an inter-intra prediction mode in which a prediction signal of the current video block is generated at least based on an intra prediction signal and an inter prediction signal, a generalized bi-prediction prediction (GBi) mode with unequal weights, a merge with motion vector differences (MMVD) mode in which a motion vector of a video block is derived based on a merge motion candidate list and is further refined by at least one motion vector difference, and an alternative temporal motion vector prediction (ATMVP) mode in which motion information of a subblock of the current video block is based on a corresponding block that is determined by motion information of a spatial neighboring block of the current video block. 2.13. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of examples 2.1 to 2.12. 2.14. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples 2.1 to 2.12. 2.15. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of examples 2.1 to 2.12. 2.16. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of examples 2.1 to 2.12. 3.1. A method for video processing, comprising: in a process of acquiring motion information of a current video block, during a conversion between the current video block and a bitstream representation of the current video block, wherein the motion information of the current video block is based on at least one affine model of at least one neighboring block of the current video block, determining whether a neighboring block of the current video block uses an Intra-Block Copy(IBC) mode in which at least one reference picture used by the neighboring video block is identical to a picture in which the neighboring block is located; disabling, for the current video block, to derive an affine candidate from the neighboring block based on determining that the neighbouring block uses the IBC mode; and performing the conversion based on the motion information of the current video block. 3.2. The method of example 3.1, wherein the conversion comprises encoding the current video block into the bitstream representation. 3.3. The method of example 3.1, wherein the conversion comprises decoding the bitstream representation of the current video block to generate pixel values of the current video block. 3.4. The method of any of examples 3.1-3.3, wherein the method further comprising: deriving, based on determining that the neighbouring block does not use the IBC mode, an affine candidate for the current video block from the neighbouring block. 3.5. The method of any of examples 3.1-3.4, wherein the affine candidate comprises an affine merge candidate, and wherein the affine candidate is in a sub-block merge candidate list 3.6. The method of example 3.5, wherein the neighbouring block is treated as unavailable based on determining that the neighbouring block uses the IBC mode. 3.7. The method of any of examples 3.1-3.4, wherein the affine candidate comprises an affine Advanced Motion Vector Prediction (AMVP) candidate, and wherein the affine candidate is in an affine AMVP list. 3.8. The method of example 3.7, wherein the neighbouring block is treated as unavailable based on determining that the neighbouring block uses the IBC mode. 3.9. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of examples 3.1 to 3.8. 3.10. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples 3.1 to

3.8. 3.11. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of examples 3.1 to 3.8. 3.12. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of examples 3.1 to 3.8. 4.1. A method for video processing, comprising: splitting a current video block into sub-blocks, during a conversion between a current video block and a bitstream representation of the current video block based on subblock-based temporal motion vector prediction (SbTMVP) mode; identifying, based on motion information of a spatial neighboring block of the current video block, a corresponding block, for a sub-block, in a motion source picture with a temporal vector; generating, based on coded mode information of the corresponding block, a motion vector and a reference index of the sub-block from the corresponding block of the sub-block; performing the conversion based on the generated motion vector and reference index. 4.2. The method of example 4.1, wherein the motion source picture is a collocated reference picture of the current video block. 4.3. The method of any of examples 4.1-4.2, wherein the corresponding block is coded with an Intra Block Copy (IBC) mode, in which a reference block of the corresponding block includes at least one pixel in a picture that includes the corresponding block. 4.4. The method of example 4.3, further comprising: treating the corresponding block as unavailable during the conversion. 4.5. The method of example 4.4, further comprising: refraining from deriving the motion vector and the reference picture of the sub-block from the corresponding block. 4.6. The method of example 4.5, further comprising: assigning default motion information to the current video block. 4.7. The method of example 4.3, wherein the corresponding block is treated as available and the motion vector and the reference picture of the sub-block is derived from the corresponding block. 4.8. The method of example 4.7, wherein the motion vector of the sub-block is copied from that associated with the corresponding block. 4.9. The method of example 4.1, wherein the spatial neighboring block is treated as unavailable if it uses an IBC mode, in which a reference block of the spatial neighboring block includes at least one pixel in a picture that includes the spatial block. 4.10. A method for video processing, comprising: splitting a current video block into sub-blocks, during a conversion between a current video block and a bitstream representation of the current video block based on subblock-based temporal motion vector prediction (SbTMVP) mode; identifying, based on coded mode information of a spatial neighboring block of the current video block, a corresponding block, for a sub-block, in a motion source picture with a temporal vector; generating a motion vector and a reference index of the sub-block from the corresponding block of the sub-block; performing the conversion based on the generated motion vector and reference index. 4.11. The method of example 4.10, wherein the motion source picture is a collocated reference picture of the current video block. 4.12. The method of any of examples 4.10-4.11, wherein the spatial neighboring block is coded with an Intra Block Copy (IBC) mode, in which a reference block of the spatial neighboring block includes at least one pixel in a picture that includes the spatial neighboring block. 4.13. The method of example 4.12, further comprising: treating the spatial neighboring block as unavailable during the conversion. 4.14. The method of example 4.13, further comprising: refraining from identifying the corresponding block based on a motion vector of the spatial neighboring block. 4.15. The method of example 4.14, further comprising: default motion information is assigned to identify the corresponding block. 4.16. The method of any examples of 4.1 to 4.15, wherein the conversion comprises encoding the current video block into the bitstream representation. 4.17. The method of any examples of 4.1 to 4.15, wherein the conversion comprises decoding the bitstream representation of the current video block to generate pixel values of the current video block. 4.18. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of examples 4.1 to 4.15. 4.19. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples 4.1 to 4.15. 4.20. A video decoding apparatus comprising a processor configured to implement a method recited in one or

more of examples 4.1 to 4.15. 4.21. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of examples 4.1 to 4.15.

(224) 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.

(225) Implementations of the subject matter and the functional operations described in the present disclosure 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.

(226) 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.

(227) 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., a field programmable gate array (FPGA) or an application specific integrated circuit (ASIC).

(228) 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., erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), and flash memory devices. The processor and the

memory can be supplemented by, or incorporated in, special purpose logic circuitry.

(229) It is intended that the specification, together with the drawings, be considered exemplary only, where exemplary means an example. As used herein, the use of "or" is intended to include "and/or", unless the context clearly indicates otherwise.

(230) While the present disclosure 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 the present disclosure 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.

(231) 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 the present disclosure should not be understood as requiring such separation in all embodiments.

(232) 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 the present disclosure.

Claims

1. A method of processing video data, comprising: determining that an intra block copy (IBC) mode is applied to a first video block of a video, wherein in the IBC mode, reference samples from a video region including the first video block are used; and performing, based on the determining, a first conversion between the first video block and a bitstream of the video, wherein the first video block is a block with a dual coding tree in which a luma component and chroma components are coded with separate separated coding structure trees, wherein the method further comprises: constructing, during a second conversion between a second video block of the video and the bitstream of the video, a motion candidate list for the second video block based on an affine motion candidate, wherein the second conversion is performed based on the motion candidate list, and wherein the affine motion candidate is derived based on a spatial neighboring block of the second video block, and wherein, in response to the spatial neighboring block being the first video block, the spatial neighboring block is excluded from deriving the affine motion candidate.

2. The method of claim 1, further comprising making a decision regarding a disabling of a specific coding mode for the first video block.

3. The method of claim 2, wherein a flag for the specific coding mode is not included in the bitstream in response to the IBC mode being used in the first video block, and wherein when the flag is not included in the bitstream, the flag is inferred to be zero.

4. The method of claim 2, wherein the specific coding mode comprises a bi-prediction with CU-level weights mode, wherein in the bi-prediction with CU-level weights mode, different weighting values relate with different reference pictures in a prediction derivation process.

5. The method of claim 4, wherein a weighting index of the bi-prediction with CU-level weights mode is not included in the bitstream in response to the IBC mode being used in the first video block, and wherein when the weighting index is not included in the bitstream, the weighting index is inferred to be 0.

6. The method of claim 2, wherein the specific coding mode comprises a merge with motion vector

difference (MMVD) mode, wherein in the MMVD mode, a motion vector of a video block is derived based on a merge motion candidate list and is further refined by at least one motion vector offset, or wherein the specific coding mode comprises an affine mode and a combined inter-intra prediction mode, wherein the affine mode uses at least one control point motion vector, and wherein in the combined inter-intra prediction mode, a final prediction is generated at least based on a weighted sum of an intermediate intra prediction signal and an intermediate inter prediction signal, or wherein the specific coding mode comprises a sub-block based temporal motion vector prediction mode, and wherein in the sub-block based temporal motion vector prediction mode, motion information is derived based on a collocated region that is determined by at least one temporal motion offset.

7. The method of claim 1, wherein before the performing, the method further comprises: deriving a block vector for the first video block; and using at least one block vector difference included in the bitstream to refine the block vector.

8. The method of claim 1, wherein a width of the first video block is greater than or equal to 2 and a height is greater than or equal to 2.

9. The method of claim 1, wherein the motion candidate list is a subblock merge candidate list.

10. The method of claim 9, wherein the spatial neighboring block is treated as unavailable based on determining that the spatial neighboring block uses the IBC mode, and wherein the subblock merge candidate list is constructed further based on a subblock-based temporal motion vector prediction candidate.

11. The method of claim 10, wherein constructing the subblock merge candidate list comprises adding of the subblock-based temporal motion vector prediction candidate is disabled in case that a temporal motion vector prediction tool is disabled or a subblock-based temporal motion vector prediction tool is disabled.

12. The method of claim 10, wherein the subblock-based temporal motion vector prediction candidate is derived by: initializing a temporal motion information to a default motion information; setting, in response to a specific neighboring block which is adjacent to a lower left corner of the second video block being available and a reference picture of the specific neighboring block being a collocated picture of the second video block, the temporal motion information to specific motion information related to the specific neighboring block; and deriving the subblock-based temporal motion vector prediction candidate based on the temporal motion information, wherein the specific neighboring block covers a luma location (xCb−1, yCb+cbHeight−1), wherein (xCb,yCb) is a luma location of a top-left sample of the second video block relative to a top-left luma sample of a current picture and cbHeight is a height of the second video block.

13. The method of claim 1, wherein the motion candidate list is a motion vector prediction candidate list, and wherein performing the second conversion based on the motion candidate list comprises deriving a motion information for the second video block based on the affine motion candidate and at least one motion vector difference which is used to refine the motion information, and wherein the spatial neighboring block is treated as unavailable based on determining that the spatial neighboring block uses the IBC mode.

14. The method of claim 1, wherein the second conversion comprises generating the bitstream from the second video block, and the method further comprises: storing the bitstream in a non-transitory computer-readable recording medium.

15. The method of claim 1, wherein the first conversion comprises decoding the first video block from the bitstream.

16. The method of claim 1, wherein the first conversion comprises encoding the first video block into the bitstream.

17. An apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to: determine that an intra block copy (IBC) mode is applied to a first video block of a

video, wherein in the IBC mode, reference samples from a video region including the first video block are used; and perform, based on the determination, a first conversion between the first video block and a bitstream of the video, wherein the first video block is a block with a dual coding tree in which a luma component and chroma components are coded with separate separated coding structure trees, wherein the instructions upon execution by the processor, further cause the processor to: construct, during a second conversion between a second video block of the video and the bitstream of the video, a motion candidate list for the second video block based on an affine motion candidate, wherein the second conversion is performed based on the motion candidate list, and wherein the affine motion candidate is derived based on a spatial neighboring block of the second video block, and wherein in response to the spatial neighboring block being the first video block, the spatial neighboring block is excluded from deriving the affine motion candidate.

18. A non-transitory computer-readable storage medium storing instructions that cause a processor to: determine that an intra block copy (IBC) mode is applied to a first video block of a video, wherein in the IBC mode, reference samples from a video region including the first video block are used; and perform, based on the determination, a first conversion between the first video block and a bitstream of the video, wherein the first video block is a block with a dual coding tree in which a luma component and chroma components are coded with separate separated coding structure trees, wherein the instructions further cause the processor to: construct, during a second conversion between a second video block of the video and the bitstream of the video, a motion candidate list for the second video block based on an affine motion candidate, wherein the second conversion is performed based on the motion candidate list, and wherein the affine motion candidate is derived based on a spatial neighboring block of the second video block, and wherein in response to the spatial neighboring block being the first video block, the spatial neighboring block is excluded from deriving the affine motion candidate.

19. A method for storing a bitstream of a video, comprising: determining that an intra block copy (IBC) mode is applied to a first video block of the video, wherein in the IBC mode, reference samples from a video region including the first video block are used; and generating the bitstream from the first video block based on the determining, storing the bitstream in a non-transitory computer-readable recording medium; wherein the first video block is a block with a dual coding tree in which a luma component and chroma components are coded with separate coding structure trees, wherein the method further comprises: constructing a motion candidate list for a second video block based on an affine motion candidate, generating the bitstream from the second video block based on the motion candidate list, and wherein the affine motion candidate is derived based on a spatial neighboring block of the second video block, and wherein in response to the spatial neighboring block being the first video block, the spatial neighboring block is excluded from deriving the affine motion candidate.